A CERTAIN CLASS OF BINARY CYCLIC BURST ERROR CORRECTION CODES

by

Peter R. McIntyre

Submitted to the Department of Electrical Engineering in partial fulfilment of the requirements for the degree of

Master of Science

Department of Electrical Engineering
Faculty of Pure and Applied Science
University of Ottawa
Ottawa, Ontario

April, 1971

ABSTRACT

This thesis is concerned with techniques for the detection and correction of binary cyclic burst errors. In particular, the thesis will introduce a class of BEC (Burst Error Correcting) Codes which are a sub-class of existing BEC codes. It will be shown that the price paid in using a sub-code in terms of rate and decoding speed is quite small relative to other codes and decoding procedures. It will also be shown that there are some advantages to using the sub-codes.

ACKNOWLEDGEMENTS

The author wishes to take this opportunity to extend his sincere gratitude to Professor S.G.S. Shiva for suggesting the topic of this thesis and for guidance in the course of development of it. His many comments and criticisms, encouragement, and time-consuming discussions were indeed valuable.

The author would also like to thank the University of Ottawa for their financial assistance, and Bell-Northern Research for the use of their computer time for the typeout of the thesis.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 CODING - GENERAL

Communication theory considers the problem of transmitting information from one point to another using a channel of some form. The two points may be very close together or very far apart. An example of the former is a modern digital or hybrid computer, where the information is transmitted from the memory unit and received by the arithmetic unit through a solid data link. Conversely, an example of the latter might be found in the Apollo space program. The transmitter is in the command module, the receiver at mission control in Houston, and the channel is the ether medium that stretches in between for one-quarter of a million miles.

Since the advent of the digital computer, digital techniques have become very popular. Computers process information digitally, and thus information is transmitted in this form. Figure 1.1 below shows a schematic diagram of a

system employing digital techniques and is called a Digital Data Transmission system. We will be using this type of system throughout in this thesis.

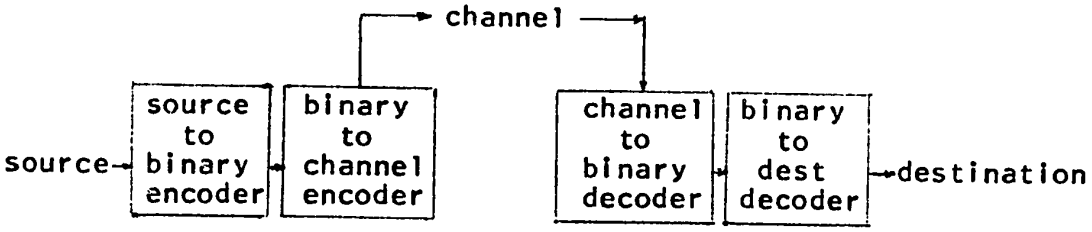


FIGURE 1.1

The source-to-binary encoder of Figure 1.1 above converts the signal source into a binary sequence (i.e. a sequence of ones and zeroes). The binary-to-channel encoder converts this sequence into a form which the channel will be able to use. For example, this 'modulation' procedure may transmit the zero and one at two different frequencies, just as automated Morse Code ham radio transmitters do. This is called frequency modulation, and other methods that may be used are phase modulation and amplitude modulation.

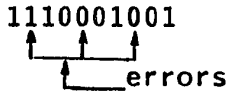
Whether the channel in Figure 1.1 is a data link or a gaseous one, there is a common problem --- NOISE. This noise could be atmospheric noise, noise due to a faulty connection,

or magnetic tape imperfections. The noise will, however, always affect the message in the same way. That is, transmission of a zero or one may, when affected by noise, result in reception of a one or zero respectively. If these bits (binary digits) are changed in a manner in which there is no correlation pattern, then we say that the errors have occurred in a random manner.

However, binary sequences are not always affected in a random manner. The most common non-random error is called a burst error. For example, consider a binary sequence transmitted as:

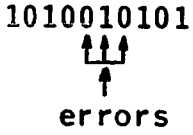
1010101101

Consider the following received sequence:

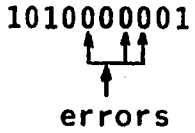


It could then be said that three errors have occurred in a

random manner (the arrows point to the bits that are in error). If however, the received sequence was;



then it can be seen that three errors have occurred in the fifth, sixth, and seventh positions. Such a sequence of errors is called a BURST ERROR of length three beginning in the fifth position. In this case, the error is more aptly called a solid burst error because the errors occurred together. Consider the following received sequence:



In this case, the error would be called a non-solid burst error of length four starting in the fifth position.

Burst errors are of particular interest because they

occur in many practical cases. A line-to-ground fault in a telephone cable caused by lightning may last long enough to erase or distort several bits in a row. Magnetic tape defects may erase more than one bit in a non-random manner. Atmospheric noise may result in non-solid burst errors occurring for a finite length of time.

Obviously then, error control is necessary to counteract the effects of noise. Generally, we use a sequence k bits long containing information, and append to it a checking sequence g bits long. The resulting sequence will be n bits long where:

$$n=g+k$$

The g checking bits will provide some information at the receiving end such that any errors within certain bounds can be corrected. The g bits are specifically designed to correct errors up to and including the maximum number of errors that are expected to occur on the average in the given channel.

Figure 1.2 below shows the Digital Data Transmission System of Figure 1.1 now with error correction and detection capability.

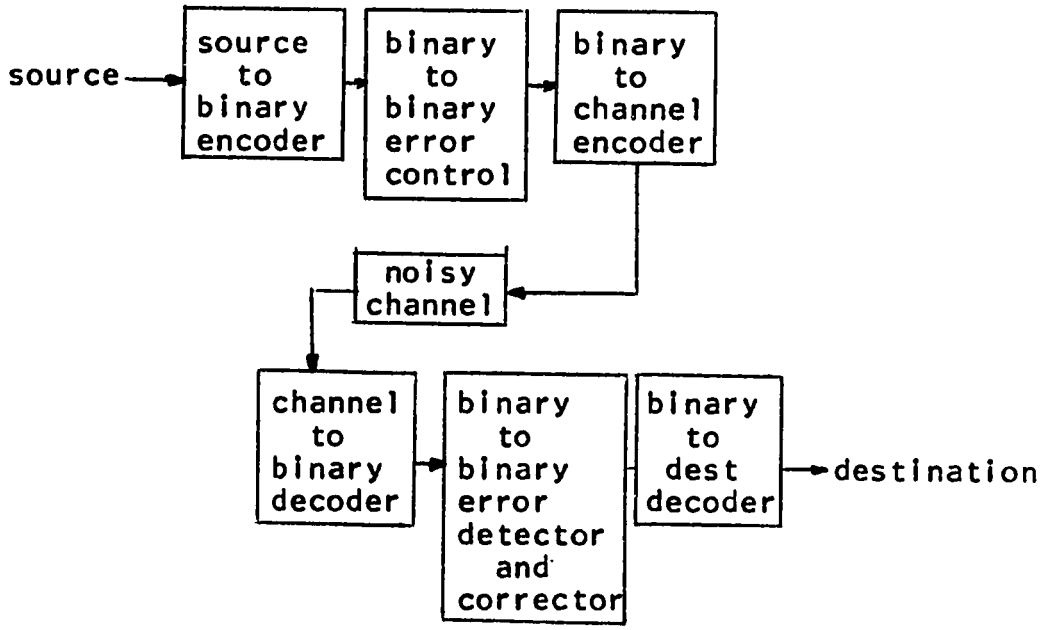


FIGURE 1.2

In Chapter Two, we will discuss some of the algebraic concepts necessary for the development of this thesis. While all are used impicitly, not all are used explicitly in the development of the methods in the following chapters.

Chapter Three will discuss coding theory with emphasis on binary cyclic codes.

Chapter Four deals with known burst error correcting codes.

Chapter Five will discuss known decoding techniques. Some of these techniques will use shift registers only and some will use shifting and off-line calculations. None,

however, uses memory storage in a computer.

Chapter Six will present a burst error correction decoding procedure which will use shifting, calculations, and memory storage. It will be shown how well this method compares with existing procedures. The main points of this procedure have also been presented as a paper to a symposium (XR20).

In Chapter Seven, certain concluding remarks will be made.

CHAPTER TWO

SOME ALGEBRAIC FUNDAMENTALS

2.1 GROUPS

A group G is a non-empty set of elements together with an operation (*) such that;

1. for all a,b in G, a*b is in G. (Closure)
2. a*(b*c) = (a*b)*c for a,b,c in G (Associativity)
3. There exists an identity element e in G such that
   $$a*e = e*a = a \quad \text{for all } a \text{ in } G$$
4. For all a in G, there exists an element a' in G called the inverse of a such that
   $$a'*a = a*a' = e$$

Note: The group is called Abelian or cummutative if
$$a*b = b*a \quad \text{for all } a,b \text{ in } G.$$

2.1.1 SUBGROUPS

A subset of elements of a group G form a subgroup H iff

leaf 9 omitted
in page numbering.

the conditions of a group are satisfied.

2.2 RINGS

A ring R is a set of elements for which two operations, addition (+) and multiplication (.), are defined such that;

1. R is an Abelian group under addition.
2. For all a,b in R, a.b is in R (closure)
3. For all a,b,c in R, a.(b.c)=(a.b).c (associativity)
4. For all a,b,c in R;
   a. a.(b+c) = a.b + a.c
   b. (b+c).a = b.a + c.a (distributivity)

Notes: 1. A ring R is commutative iff
a.b = b.a for all a,b in R
2. A ring with identity e under multiplication is called a 'ring with identity'

2.2.1 SUBRINGS

A subset of the elements of R form a subring A iff;

1. A is also a ring, and
2. e is in A.

2.3 FIELDS

A field F is a commutative ring with identity in which every non-zero element has an inverse under multiplication. For example, the real numbers have all the characteristics of a ring, have an identity under multiplication (1), and every element has an inverse. Thus, it follows that the real numbers form a field.

2.4 VECTOR SPACES

A set V is a vector space over a field F iff;

1. V is an Abelian group under addition
2. For any vector v in V, and any scalar c in F, a vector cv in V is defined
3. For all u,v in V, and a,b in F;
   a. $a(u+v) = au + av$
   b. $(a+b)u = au + bu$ (Distributivity)
4. For all v in V, and a,b in F;
   a. $(a.b)v = a(bv)$
   b. $ev = v$ (Associativity)

2.4.1 VECTOR SUBSPACES

A subset of V is called a vector subspace iff it satisfies all the conditions of a vector space.

2.4.2 LINEAR INDEPENDENCE OF VECTORS

Consider a set of vectors v_i, $i=1,\ldots,k$ in V. These vectors are said to be linearly independent iff

$$\sum_{i=1}^{k} c_i v_i = 0$$

$$\Longrightarrow c_i = 0 \text{ for all } i=1,\ldots,k,\ c_i \text{ in } f$$

Otherwise, the vectors are said to be linearly dependent.

Example: Let $v_1 = (1,0,1,1)$

$v_2 = (0,1,1,0)$

$v_3 = (1,0,0,0)$

Then,

$$c_1v_1 + c_2v_2 + c_3v_3 = (c_1+c_3, c_2, c_1+c_2, c_1)$$

$$= (0,0,0,0) \text{ iff}$$

$c_1 + c_3 = 0$, $c_2 = 0$, $c_1 + c_2 = 0$, and $c_1 = 0$.

i.e. $c_1 = c_2 = c_3 = 0$

Thus, v_1, v_2, and v_3 are linearly independent.

2.5 OPERATIONS ON BINARY SEQUENCES

2.5.1 MODULO-TWO ADDITION

If $a_1, a_2, a_3, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ are two binary sequences each of length n, then their modulo-2 sum is a third sequence $c_1, c_2, \ldots, c_n$, also of length n where $c_i = a_i + b_i$ and + indicates modulo-2 addition defined by

1. $1+1 = 0+0 = 0$
2. $0+1 = 1+0 = 1$

2.5.2 WEIGHT

Let $A = (a_1, a_2, \ldots, a_n)$ be a binary sequence of length n (i.e. an n-tuple). Then the weight of A, denoted W(A), is the number of ones in A.

2.5.3 HAMMING DISTANCE

The Hamming distance d between two binary n-tuples A and B is defined by

$$d = W(A + B)$$

Example: $A = (1,0,1,1,0,1,1)$

$B = (0,1,1,1,0,1,0)$

$A + B = (1,1,0,0,0,0,1)$

and thus $d = 3$

It may be seen that another definition could be that the Hamming distance between two binary n-tuples is the number of places in which they differ.

2.6 POLYNOMIALS

2.6.1 GENERAL

A polynomial is a function of a variable over a field and is related to a sequence. In this thesis, we will deal with sequences and polynomials in the following manner. If a sequence A is defined by

$$A = (a_1, a_2, \ldots, a_n),$$

then the corresponding polynomial is

$$A(X) = a_1 + a_2X + a_3X^2 + \ldots + a_nX^{n-1}.$$

Example: If $A = (1,0,0,1,1,0,1)$, the polynomial representation would be

$$A(X) = 1 + 0.X + 0.X^2 + 1.X^3 + 1.X^4 + 0.X^5 + 1.X^6$$
$$= 1 + X^3 + X^4 + X^6$$

2.6.2 ADDITION (MODULO-TWO)

Addition is done in the ordinary manner with the exception that modulo-2 arithmetic is implemented.

EXAMPLE: If $A(X) = 1 + X^2 + X^3$, and

$B(X) = X + X^2$, then

$C(X) = A(X) + B(X)$

$= 1 + X + X^2(1+1) + X^3$

$= 1 + X + X^3$.

2.6.3 MULTIPLICATION

Let $A(X) = a_0 + a_1X + \ldots + a_{n-1}X^{n-1}$, and

$B(X) = b_0 + b_1X + \ldots + b_{n-1}X^{n-1}$ be two polynomials.

Define their product C(X) by

$$C(X) = c_0 + c_1X + \ldots + c_{2(n-1)}X^{2(n-1)}, \text{ where}$$

$$c_k = \sum_{i=0}^{k} a_i b_{k-i}$$

2.6.4 MONIC POLYNOMIALS

Let $f(x) = a_0 + a_1X + \ldots + a_nX^n$. The polynomial f(x) is said to be monic if $a_n=1$.

2.6.5 IRREDUCIBLE POLYNOMIALS

Consider a polynomial f(x) over a field F. f(x) is called irreducible if, whenever some h(x) divides f(x), either h(x) = c or h(x) = cf(x), where c is a scalar. Example: $f(x) = 1 + X + X^2$ is irreducible over the binary field F(0,1).

2.7 IDEALS

2.7.1 DEFINITION

An ideal G is a subgroup of a ring R iff

1. G is a subset of the additive group of R.
2. For any g in G, and r in R, gr and rg are in G.

EXAMPLE: A code word, generated by a 'generator polynomial' as discussed in Chapter Three, and its cyclic shifts form an ideal.

2.7.2 POLYNOMIALS

A set of polynomials is an ideal iff it consists of all

multiples of some polynomials. In future, this set will be denoted $\{p(X)\}$.

2.8 RESIDUE CLASSES

A residue class is a set of polynomials of the form $a(X) + \{p(X)\}$ where $a(X)$ is in $F(X)$.

2.9 MINIMUM POLYNOMIAL

Let β be an element of an extension field over F. The monic polynomial $m(X)$ in F of smallest degree such that $m(b)=0$ is called the minimum polynomial of b.

The minimum polynomial $m(x)$ of any element b in an extension field is irreducible.

2.10 FINITE FIELDS

A finite field is a field with a finite number of elements.

2.10.1 GALOIS FIELDS

Let p be a prime integer and n a positive integer. Then

there exists a field with p^n elements called a Galois field and denoted by $GF(p^n)$.

2.10.2 MULTIPLICATIVE GROUP OF A GALOIS FIELD

Consider in any finite group the set of elements formed by any element α and all its powers α^2, α^3, et cetera. There can be at most a finite set of such elements and therefore at some point there must be repetition. i.e. $\alpha^i = \alpha^j$ for some distinct i and j. Post-multiplication by α^{-i} gives

$$\alpha^i\alpha^{-i} = \alpha^j\alpha^{-i}$$

or,

$$\alpha^{j-i} = 1.$$

Let e be the smallest integer such that $\alpha^e = 1$. Then e is called the order of the element α. The set $1,\alpha,\ldots,\alpha^{e-1}$ forms a multiplicative subgroup. The group that consists of all the powers of one of its elements is called a cyclic group.

In the Galois field of q elements, GF(q), there is a primitive element α, i.e. an element of order q-1, such that every non-zero element can be expressed as a power of α. This is called the multiplicative cyclic group GG(q).

Example: The Galois field of 16 elements, $GF(2^4)$, may be

formed as a field of polynomials over GF(2) modulo $1+X+X^4$. Let α denote the residue class containing X. α is a root of $1+X+X^4$ and it is a primitive element of that field. The non-zero elements of the field are given below in Table 2.1.

$\alpha^0 = 1$

$\alpha^1 = \alpha$

$\alpha^2 = \alpha^2$

$\alpha^3 = \alpha^3$

$\alpha^4 = 1 + \alpha$

$\alpha^5 = \alpha + \alpha^2$

$\alpha^6 = \alpha^2 + \alpha^3$

$\alpha^7 = 1 + \alpha + \alpha^3$

$\alpha^8 = 1 + \alpha^2$

$\alpha^9 = \alpha + \alpha^3$

$\alpha^{10} = 1 + \alpha + \alpha^2$

$\alpha^{11} = \alpha + \alpha^2 + \alpha^3$

$\alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$

$\alpha^{13} = 1 + \alpha^2 + \alpha^3$

$\alpha^{14} = 1 + \alpha^3$

$\alpha^{15} = 1 = \alpha^0$

TABLE 2.1

FIELD ELEMENT REPRESENTATION OF GF(2^4)

GENERATED BY $1 + X + X^4$

CHAPTER THREE

BINARY CYCLIC CODES

3.1 DEFINITION

An (n,k) linear code V is said to be cyclic if, for every V(X) in V, ($X^i V(X)$ modulo $1+X^n$) is also in V, where n is the code length and k is the number of information bits.

Consider the ideal generated by g(X), where g(X) divides $1+X^n$. If P(X) is a polynomial in this ideal, the following relation holds:

$$X^i P(X) = (1+X^n)A(X) + Q(X)$$

where Q(X) is the remainder. That is,

$$X^i g(X) I(X) = (1+X^n)A(X) + Q(X)$$

where I(X) is the information polynomial.

Since g(X) divides $1+X^n$, then

$$(1+X^n)A(X) \text{ modulo } g(X) = 0,$$

and thus

$$Q(X) \text{ modulo } g(X) = 0.$$

In other words, Q(X) belongs to the ideal. Thus the ideal generated by g(X) is a cyclic code.

3.2 PARITY CHECK MATRIX

For the code previously described and generated by g(X) of degree n-k, define the parity check polynomial $h^*(X)$ by

$$h^*(X) = (1+X^n)/g^*(X)$$

where $f^*(X)$, the reciprocal of f(X) is defined by

$$f^*(X) = X^a f(1/X)$$

and where a is the degree of f(X).

The n-k by n matrix H with $h^*(X)$ and its n-k-1 cyclic shifts is a parity check matrix for V_j; that is, V is the null space of H.

3.3 BOSE-CHAUDHURI-HOCQUENGHEM CODES

The method of construction of these codes (XR3,XR8) known as BCH codes is as follows:

Let c be the smallest number such that nc can be expressed in the form 2^m-1. Then a generator polynomial for a BCH code that can correct up to t errors (an (n,k,t) BCH code) is defined by

$$g(x) = \text{LCM } (g_1(X), g_3(X), g_5(X), \ldots, g_{2t-1}(X))$$

where $g_i(X)$ is the minimal polynomial of α^{ic}, α being a primitive element of $GF(2^m)$.

3.3.1 PRIMITIVE CODES

The BCH codes for which c=1 are called primitive codes.

Conversely, if c is not 1, the code is called non-primitive.

3.3.2 DECODING PROCEDURES

Decoding of BCH codes may be done in many ways, but the most accepted method is the Peterson procedure (XR13).

BCH codes are the largest class of random error correcting codes. For a more detailed treatment of cyclic codes in general, and BCH codes in particular, reference should be made to Peterson (XR12).

3.3.3 EXAMPLE

Consider the code with m=4, t=2, c=1. The generator polynomial then has the form

$$g(X) = g_1(X)g_3(X)$$

where: 1. g(x) divides $1+X^n$, and $n=2^4-1=15$.

2. $g_1(X)$ is a primitive polynomial with root α such that α is a primitive element of GF(16).

3. $g_3(x)$ is a minimal polynomial with root α^3.

We know that over GF(16), $\alpha^4=1+\alpha$, and thus

$$g_1(X) = 1+X+X^4.$$

Assume that $g_3(X)$ has root α^3 and is of the form

$$g_3(X) = \beta_0+\beta_1X+\beta_2X^2+\beta_3X^3+\beta_4X^4.$$

Using Table 2.1 and solving for $g_3(\alpha^3) = 0$, one gets

$$\beta_0 = \beta_1 = \beta_2 = \beta_3 = \beta_4 = 1.$$

Thus, $g(X) = g_1(X)g_3(X)$

$$= (1+X+X^4)(1+X+X^2+X^3+X^4)$$

$$= 1+X^4+X^6+X^7+X^8$$

Also, $g = n-k = 8$, and $n = 15$, thus $k = 7$, and we have a (15,7,2) BCH code generated by $g(X) = 1+X^4+X^6+X^7+X^8$. Further, $g^*(X) = 1+X+X^2+X^4+X^8$.

Thus, $h^*(X) = (1+X^{15})/g^*(X)$

$$= 1+X+X^3+X^7$$

and,

$$H = \begin{matrix} 110100010000000 \\ 011010001000000 \\ 001101000100000 \\ 000110100010000 \\ 000011010001000 \\ 000001101000100 \\ 000000110100010 \\ 000000011010001 \end{matrix}$$

H is an 8 x 15 or n-k x n matrix with rank n-k.

CHAPTER FOUR

BURST ERROR CORRECTING CODES

4.1 INTRODUCTORY THEORY

In Chapter One, the concepts of solid and non-solid burst errors were introduced in general terms. This chapter will discuss the codes that will correct such errors. Such codes are called burst error correcting codes (BEC codes).

If a burst error occurs, the error polynomial will be of the form

$$E(X) = X^{i}B(X)$$

where B(X) is a polynomial of finite degree (b-1). That is, the length of the burst is b, and B(X) is of the form

$$B(X) = 1+a_1X+a_2X^2+\ldots+a_{b-2}X^{b-2}+X^{b-1}.$$

For example, a burst of length 5 might have

$$B(X) = 1+X+X^2+X^4 \text{ (non-solid)},$$

and thus, if the burst started in the tenth position,

$$E(X) = X^9(1+X+X^2+X^4).$$

4.2 ABRAMSON CODES

Abramson (XR1) has described a class of codes which can correct all bursts of length 2 or less, and another class which can correct all bursts of length 3 or less.

4.2.1. CODES FOR $b \leqslant 2$

Abramson proved that a generator polynomial

$$g(x) = P(X)(1+X)$$

where P(X) is a primitive polynomial of degree q and the length of the code is $n=2^q-1$, could correct bursts of length 2 or less.

THEOREM 4.1: Consider a code generated by $g(X)=P(X)(1+X)$ with $n=2^q-1$ and P(X) is a primitive polynomial of degree q. Then this code will correct bursts of length 2 or less.

Proof:

$$g(X) = P(X)(1+X)$$

Then any codeword V(X) would have the form

$$V(X) = P(X)(1+X)I(X)$$

where I(X) has degree $\leqslant (n-1)-(q+1)=n-q-2$. The received polynomial, affected by error, would be

$$R(X) = P(X)(1+X)I(X) + E(X)$$

where E(X) could be

1. 0 if there is no error,
2. X^i if a single error,
3. $X^j(1+X)$ if a double adjacent error.

Consider the syndromes of R(X), namely $S_k(X)$, where

$$S_0(X) = P(X)(1+X)I(X) \qquad E(X)=0$$

$$S_1(X) = P(X)(1+X)I(X) + X^i \qquad E(X)=X^i$$

$$S_2(X) = P(X)(1+X)I(X) + X^j(1+X) \qquad E(X)=X^j(1+X)$$

To prove the theorem, it is necessary to prove only that the syndromes are distinct for all roots of the generator polynomial. That is, for $X=\alpha$ and $X=1$.

Case 1: $X = \alpha$.

$$S_0(\alpha) = 0$$

$$S_1(\alpha) = \alpha^i$$

$$S_2(\alpha) = \alpha^j(1+\alpha)$$

Now, consider some $\alpha^k = 1 + \alpha$. Then,

$$S_0(\alpha) = 0$$

$$S_1(\alpha) = \alpha^i$$

$$S_2(\alpha) = \alpha^{j+k}$$

Certainly then, the syndromes S_1 and S_2 for $X=\alpha$ are

distinct except when $i=(j+k) \bmod n$. S_0 is always different from S_1 and S_2.

Case 2: $X = 1$.

$$S_0(1) = 0$$

$$S_1(1) = 1$$

$$S_2(1) = 0$$

Here, S_1 and S_2 are different. Case 1 indicated that S_1 and S_2 could be the same, but case 2 pre-empts that possibility, and thus S_1 and S_2 are distinguishable by the code.

Therefore, $g(X) = P(X)(1+X)$ can distinguish between errors for b=0,1, and 2, and can correct them.

The factor (1+X) adds an extra parity check on all digits. Thus g(x) has degree q+1, and

$$k+q+1 \leqslant n$$

$$k+q+1 \leqslant 2^q-1$$

$$k \leqslant 2^q-q-2$$

The rate of this code is

$$R = (2^q-q-2)/(2^q-1)$$

es ? and the rate approaches unity as either q or n approach infinity.

DECODING:

$$V(X) = P(X)(1+X)$$

which always has even weight. Since

$$R(X) = V(X) + E(X),$$

the weight of R(X) depends on the weight of E(X). Further,

$$R(\alpha) = E(\alpha).$$

If R(X) has odd weight, then E(X) will also have odd weight; that is, $E(X) = X^i$ and $E(\alpha) = \alpha^i$. Since all α^i's are distinct, we can find i by computing R(α).

If R(X) has even weight, then E(X) has even weight; that is,

1. $E(X) = 0$,

or 2. $E(X) = X^i(1+X)$.

Thus,

1. $R(\alpha) = 0$,

or 2. $R(\alpha) = \alpha^i(1+\alpha)$.

But $R(\alpha) = 0$ implies that $E(X)=0$ and thus if R(X) has even weight and R(α) is non-zero, then we know that b=2.

4.2.2 CODES FOR $b \leq 3$

Abramson derived such a set of codes described by the generator polynomial

$$g(X) = (1+X+X^2)P(X)$$

where P(X) is a primitive polynomial of even degree q, $n=2^q-1$. He conjectures that there is at least one P(X) for every even $q \geq 4$ such that g(X) is the generator of a three error correcting code. Some generator polynomials that may be used are in Table 4.1.

q	g(X)
4	$(1+X+X^2)(1+X+X^4)$
6	$(1+X+X^2)(1+X+X^6)$
6	$(1+X+X^2)(1+X+X^2+X^5+X^6)$
8	$(1+X+X^2)(1+X+X^2+X^7+X^8)$
8	$(1+X+X^2)(1+X+X^3+X^5+X^8)$

SOME GENERATOR POLYNOMIALS FOR $b \leq 3$ BEC CODES

TABLE 4.1

g(X) will have degree q+2 and thus k=n-(q+2), and the rate for this code will be

$$R = (n-q-2)/n$$

$$R = 1 - (q+2)/(2^q-1)$$

As with the two error correcting code, the rate approaches

unity as n approaches infinity, but the rate of approach is slower. That is, the two error correcting code is slightly more efficient. For example, if q=6,

$$R_2 = 56/63 = 0.889,$$

while $$R_3 = 55/63 = 0.873.$$

4.3 FIRE CODES

The Fire codes (XR6) are a large class of codes which are cyclic and thus can best be described in terms of their generator polynomial which has the form

$$g(X) = P(X)(1+X^c)$$

where

1. P(X) is an irreducible polynomial of degree q with root α such that $\alpha^e=1$,
2. c is not divisible by e,

and 3. n = LCM (e,c).

The number of parity checks is c+q. Thus,

$$k = n - (c+q)$$

and the rate R is

$$R = (n-c-q)/n$$

$$= 1 - (c+q)/n$$

Once again, the rate approaches 1 as n approaches infinity.

If d is the maximum burst length that the code can

correct, then Peterson (XR14) proves that the following conditions must hold:

1. $d \geq b$,
2. $b \leq q$,

and 3. $c \geq b+d-1$.

EXAMPLE: Suppose we want b=d=3 or less and of comparible length to the Abramson code example. That is, n should be about 63.

$$n = \text{LCM}(e,c)$$

and,

$$c \geq 3+3-1 = 5$$

Then, let us choose c=6 and e=7, and thus

$$n = 6 \times 7 = 42.$$

The generator polynomial for this code would be

$$g(X) = (1+X^6)(1+X+X^3).$$

The rate of the code would be

$$R = (42-9)/42 = 0.786.$$

The shortened Abramson code (n=42) has the rate

$$R = (42-8)/42 = 0.810.$$

Thus, it may be seen that the Abramson code in this case

is slightly more efficient.

If we wish to have a more efficient code, the best way to do it is to choose c such that e is a multiple of c. i.e.

$$e = kc.$$

In this case, $R = 1 - (c+q)/e$

$$= 1 - (c+q)/kc$$

$$= 1 - 1/k - q/kc$$

As n approaches infinity, R approaches (k-1)/k.

That is, R will still approach 1, but only if e is much larger than c, and slowly at that.

EXAMPLE: Consider a similar example to the previous one, but choose c=7 and e=63. i.e.

$$n = \text{LCM}\ (7,63) = 63$$

and

$$g(X) = (1+X^6)(1+X+X^3)$$

with q=6.

The rate of the code would then be

$$R = 1 - (7+6)/63 = 0.794$$

It is interesting to note that the Fire codes have even weight since (1+X) is a factor of $(1+X^c)$. Thus, if the weight of R(X) is even or odd, then there will be an even or odd number of errors respectively.

4.4 CHIEN CODES

Chien (XR5) has developed some excellent BEC codes that will detect and correct all bursts of a specified length. They are cyclic, and thus are best described by their generator polynomial

$$g(X) = (1+X^c)\prod_{j=1}^{m} P_j(X)$$

where 1. each $P_j(X)$ is a distinct irreducible polynomial of degree δ_j and order e_j,

and 2. the e_j's do not divide c.

Given the above conditions, it follows that g(X) has no repeated factors. The code length then, is

$$n = \text{LCM}\ (c, e_1, e_2, \ldots, e_m)$$

Chien has proven that this code will detect all error bursts of length $\leq$ d and will correct all bursts of length $\leq$ b that are relatively prime to $\prod_{j=1}^{m} P_j(X)$, provided that $c \geq b+d-1$ and $b \leq \sum_{j=1}^{m} \delta_j$.

EXAMPLE: Let m=2, c=11, and

$$g(X) = (1+X^{11})(1+X+X^4)(1+X+X^3)$$

Now, $\delta_1=4$, $\delta_2=3$, and $\sum_{j=1}^{2} \delta_j=7$. That is, $b \leqslant 7$. Also, $c \geqslant b+d-1$, or

$$b+d-1 \leqslant 11$$

$$b+d \leqslant 12$$

If b=d, then

$$b = 6 < 7$$

and

$$n = \text{LCM}\ (11,15,7) = 1155$$

The above example seems to have an unnecessarily long code length. However, Chien shows that with the proper use of shift register configurations and a few calculations, the code may be decoded in 30 shift register cycles (very fast). A shorter Fire code that will correct one burst of length 6 and has generator polynomial

$$g(X) = (1+X^{11})(1+X+X^4)$$

has length of only 693 but requires about 40 cycles to decode using Chien's configuration and conventionally 693 cycles.

The number of check digits is $c+\delta_1+\delta_2+\ldots+\delta_m+1$. Thus,

$$k = n - (1+c+\sum_{j=1}^{m} \delta_j)$$

and,

$$R = 1 - (1+c+\sum_{j=1}^{m} \delta_j)/\text{LCM}\ (c, e_1, e_2, \ldots, e_m).$$

For the example above, n=1155, and

$$R = 1 - (1+11+4+3)/1155 = 0.984$$

This, of course, is an excellent rate!

In Table 4.2 and Table 4.3, some generator polynomials and their error correcting abilities are shown. The symbols used are:

1. n = the code length
2. s = number of cycles needed in high speed decoding
3. b_0 = the maximum length for which all lengths are corrected
4. b = the maximum length for which most lengths are corrected
5. E = the percentage of bursts of length $> b_0$ but $\leqslant$ b that are corrected by the code
6. R = code rate

All bursts of length $\leqslant$ b are detected.

g(X)	n	s	b_0	b	E	R
$(1+X^{11})(1+X+X^4)(1+X+X^3)$	1155	30	3	6	79	0.984
$(1+X^{13})(1+X+X^4)(1+X+X^3)$	1365	31	3	7	79	0.985
$(1+X^{13})(1+X^2+X^5)(1+X+X^3)$	2821	49	3	7	82	0.992
$(1+X^{15})(1+X^2+X^5)(1+X+X^3)$	3255	52	3	8	82	0.993
$(1+X^{17})(1+X^2+X^5)(1+X+X^4)$	7905	55	4	9	90	0.997
$(1+X^{19})(1+X+X^6)(1+X^2+X^5)$	37107	90	5	10	95	0.999
$(1+X^{21})(1+X^3+X^7)(1+X^2+X^5)$	82677	157	5	11	96	1.000

CHIEN CODES FOR m=2

TABLE 4.2

g(X)	n	s	b_0	b	E	R
$(1+X^{17})(1+X^2+X^5)(1+X+X^3)(1+X+X^2)$	11067	55	2	9	49	0.997
$(1+X^{17})(1+X^2+X^5)(1+X+X^4)(1+X+X^3)$	55335	55	3	9	75	0.999
$(1+X^{19})(1+X^2+X^5)(1+X+X^3)(1+X+X^2)$	12369	58	2	10	49	0.998
$(1+X^{19})(1+X^2+X^5)(1+X+X^4)(1+X+X^3)$	61845	58	3	10	75	0.999
$(1+X^{21})(1+X^3+X^7)(1+X^2+X^5)(1+X+X^3)$	578739	157	3	11	81	1.000
$(1+X^{23})(1+X^2+X^5)(1+X+X^4)(1+X+X^3)$	74865	64	3	12	75	1.000
$(1+X^{23})(1+X^3+X^7)(1+X+X^4)(1+X+X^3)$	306705	160	3	12	78	1.000
$(1+X^{23})(1+X^3+X^7)(1+X^2+X^5)(1+X+X^3)$	633857	160	3	12	81	1.000

CHIEN CODES FOR m=3

TABLE 4.3

4.5 A CERTAIN CLASS OF CODES

Gorog (XR7) postulated several classes of BEC codes which are called systematic, perfect, and efficacious. A quick review of their forms and rates is presented here.

4.5.1 SYSTEMATIC CODES

The systematic codes are described by their generator polynomial as

$$g(X) = P_1(X)P_2(X)$$

where 1. $P_1(X)$ is irreducible, of degree q_1, and of order e_1,

and 2. $P_2(X)$ is a primitive polynomial of degree q_2 which does not divide $P_1(X)$.

If $q_2 \geq b$, and $2^{q_2}-1$ is relatively prime to e_1, then the code will correct any burst of length b in a message length

$$n = (2^{q_2}-1)L$$

where L is the length of the shortest cycle of $P_1(X)$.

Some polynomials which have the properties of $P_1(X)$ are the following:

$F(X) = X^{2b-1}+1$ $\qquad L_F = 2b-1$

$G(X) = X^{2b-2}+1$ $\qquad L_G = b-1$

$H(X) = X^{2b-1}+X^{2b-2}+\ldots+X+1$ $\qquad L_H = b$

$I(X) = X^{2b}+X^{b}+1$ $\qquad L_I = 3b$

$J(X) = X^{2b}+X^{2b-1}+\ldots+X+1$ $\quad L_J = 2b+1$

$K(X) = X^{2b}+X^{2b-2}+\ldots+X^2+1$ $\quad L_K = 2b+2$ if b even

$\quad = b+1$ if b odd

$A(X) = X+1$ $\quad L_A = 1$

It may be noted that F(X) and A(X) are the Fire code and Abramson code cases. For example, if b=3,

$$F(X) = X^5+1 = P_1(X)$$

$$P_2(X) = 1+X+X^6 \text{ say,}$$

and thus,

$$g(X) = (1+X+X^6)(1+X^5)$$

as in sec 4.3 with rate R = 0.968.

It is of interest to compare the rates of the different codes. This is done in Table 4.4.

TYPE	n	RATE
R_F	$(2b-1)(2^{q_2}-1)$	$1-(2b+q_2-1)/(2^{q_2}-1)(2b-1)$
R_G	$(b-1)(2^{q_2}-1)$	$1-(2b-2+q_2)/(2^{q_2}-1)(b-1)$
R_H	$b(2^{q_2}-1)$	$1-(2b-1+q_2)/b(2^{q_2}-1)$
R_I	$3b(2^{q_2}-1)$	$1-(2b+q_2)/3b(2^{q_2}-1)$
R_J	$(2b+1)(2^{q_2}-1)$	$1-(2b+q_2)/(2b+1)(2^{q_2}-1)$
R_K	$(b+1)(2^{q_2}-1)$	$1-(2b+q_2)/(2^{q_2}-1)(b+1)$...b odd
	$2(b+1)(2^{q_2}-1)$	$1-(2b+q_2)/2(2^{q_2}-1)(b+1)$...b even
R_A	$2^{q_2}-1$	$1-(1+q_2)/(2^{q_2}-1)$...b=2 only

TABLE 4.4 CODE RATES OF THE SYSTEMATIC CODES

By making rate comparisons for comparable lengths, it may be seen that R_G is the best systematic code. However, the Fire codes with rate R_F have almost as good a rate for a given length.

4.5.2 PERFECT CODES

For the best rate, the systematic codes must be long. The perfect codes have shorter lengths for comparable rates.

Consider a polynomial $P_A(X)$ with period N. Suppose that

there are s cycles of length N. Thus

$$n = sN$$

and the code will correct a burst of b errors in a message of length N if

$$2^{b-1} < s < 2^{b}.$$

Consider a second class of codes $P_B(X)$ given by the product of different irreducible non-primitive polynomials of the same period.

EXAMPLE: b=7, n-k=16, n=51

$$g(X) = (1+X+X^4+X^5+X^6+X^7+X^8)(1+X+X^3+X^4+X^8)$$

and,

$$P_B(X) = g(X)I(X)$$

A third class of perfect codes $P_C(X)$ is a shortened cyclic code. That is, perfect codes which correct any burst of b errors in a message of length n will correct any burst of d+1 errors in a message of length $n_1<n$.

A final class is called the class of good codes. These are formed by multiplying a perfect code of message length n_1 by a new primitive polynomial of period n_2 relatively prime to n_1, and shortening the message such that bursts are corrected.

EXAMPLE: b=4, n-k=14, and $g(X)=P_1(X)P_2(X)$

where 1. $P_1(X) = 1+X+X^2+X^4+X^9$,

and 2. $P_2(X) = 1+X^2+X^5$.

Polynomial $P_1(X)$ corrects any burst of length 3 in a message of length 73. Thus n =73. $P_2(X)$ is primitive, and thus $n_2=31$. Therefore,

$$n = n_1 n_2$$
$$= 73 \times 31$$
$$= 2263$$

Gorog shows that, to correct 4 errors, the code must be shortened to 1045.

4.5.3 EFFICACIOUS CODES

Gorog presents several classes of these codes, but one is most efficient and thus of most interest. This code is represented by $E_0(X)$ and was originally discovered by Melas (XR9).

Let $P_1(X)$ be a polynomial of degree q_1 and period N_1, and let $P_2(X)$ be a primitive polynomial of degree $q_2>q_1$ and period N_2 such that:

1. N_1 divides N_2,

and 2. $P_1(X)$ and $P_2(X)$ are relatively prime.

Then, $g(X)=P_1(X)P_2(X)$ generates a code that can correct any

burst of (q_1+1) errors in a message length of N_2.

EXAMPLE: $P_1(X) = 1+X+X^2$, $q_1=2$, $N_1=3$

$P_2(X) = 1+X+X^4$, $q_2=4$, $N_2=15$

Thus,

$$g(X) = (1+X+X^2)(1+X+X^4), \quad b=3, \quad n=15$$

The rate of this b=3 code is

$$R = (15-6)/15 = 0.600$$

4.6 MULTIPLE BEC USING RANDOM ERROR CORRECTING CYCLIC CODES

Tavares and Shiva (XR19). Consider an (n,k) binary cyclic code C with minimum distance d. If E(X) represents a combination of b burst errors, each of length T_i, $i=1,\ldots,b$, then the total weight of the b bursts would be

$$W(E(X)) = \sum_{i=1}^{b} T_i - m$$

where m is the total of the missing terms in non-solid bursts. If

$$W(E(X)) < d,$$

then E(X) cannot be a code word in C. i.e.

$$\sum_{i=1}^{b} T_i - m < d \qquad \ldots\ldots A$$

Similarly, it may be shown easily that

$$W(E(X)(1+X)) \leqslant 2b+2m$$

and if E(X)(1+X) is not in C, then neither is E(X).

Thus, $2b+2m < d$

or $m < d/2 - b$ .....B

Substituting B in A gives

$$\sum_{i=1}^{b} T_i < d+d/2-b = (3d-2b)/2$$

or,

$$\sum_{i=1}^{b} T_i \leqslant (3d-2b-1)/2 \quad .....C$$

If $b < (d-1)$, then

$$\sum_{i=1}^{b} T_i \leqslant \text{Max} \left(\left[(3d-2b-1)/2 \right], d-1 \right) \quad .....D$$

where the [ ] brackets mean the greatest whole integer.

Similarly, while D is a detection inequality, a correction inequality may be derived giving

$$\sum_{i=1}^{b} T_i < \text{Max} \left(\left[(3d-4p-1)/4 \right], \left[(d-1)/2 \right] \right) \quad .....E$$

where p is the number of bursts to be simultaneously corrected, and b=2p.

It may be noted that there exists a decoding procedure.

4.7 MULTIPLE BEC CODES

Bahl and Chien (XR2) postulated a class of cyclic codes that could correct multiple bursts of errors. Briefly, their codes are constructed as follows:

A single parity check code may be considered as a cyclic code having the generator polynomial

$$g(X) = 1+X$$

Let n_i, $i=1,\ldots,p$ be pairwise relatively prime integers such that

$$n_1 < n_2 < \ldots < n_p.$$

Then the code C of length $n=\prod_{i=1}^{p} n_i$ is constructed by forming the product of p single parity check codes c_i having block lengths $n_1, n_2, \ldots, n_p$ respectively.

i.e.
$$n = \prod_{i=1}^{p} n_i$$
$$k = \prod_{i=1}^{p} (n_i - 1)$$
and
$$g(X) = \mathrm{LCM}(1+X^{m_1}, \ldots, 1+X^{m_p})$$

where $m_i = n/n_i$.

Bahl and Chien derive two major results:

1. If $t=\lfloor p/2 \rfloor$, where t is the number of bursts of length b_t that can be corrected, then
$$b_t = \prod_{i=1}^{p-2} n_i$$
2. For $p \geq 3$, C can correct all burst patterns consisting of 2^{p-2} or fewer bursts of length n_1.

EXAMPLES:

1. Consider $n_1=5, n_2=6, n_3=7, n_4=11$.

   Therefore, $p=4$, $t=2$, $b_t=5\times6=30$, $n=5\times6\times7\times11=2310$, and $k=4\times5\times6\times10=1200$.

   Thus we have a (2310,1200) code that can correct

up to 2 bursts of length 30 or less, and generated by

$$g(X) = LCM(1+X^{462}, 1+X^{385}, 1+X^{330}, 1+X^{210})$$

The rate of this code is R = 1200/2310 = 0.519.

2. Using the same data as in 1 with $p=4 \geq 3$. The code will then correct, with the same generator polynomial, four bursts of length five.

3. Bahl and Chien also pointed out that this code will correct a single burst of length 77.

The rate of the code C is

$$R = \prod_{i=1}^{p}(n_i-1)/n_i$$
$$= \prod_{i=1}^{p}(1-1/n_i)$$
$$\leq (1-1/n_1)$$

Thus, if n_1 is small, the code will have a very poor rate. However, for a large n_1, an excellent rate is possible.

EXAMPLE: If $n_1=37$, $n_2=41$, $n_3=43$, $n_4=47$, the rate is 0.907.

4.8 CYCLIC PRODUCT CODES

Burton and Weldon (XR4). In general, the direct product of two cyclic codes may not be cyclic. However, if the block lengths are relatively prime and the digits are ordered cyclically, then the product code is cyclic.

Consider two cyclic codes A and B of lengths n_A and n_B and generated by $g_A(X)$ and $g_B(X)$ respectively. Let C be the product code formed by the operation

$$C = A \times B$$

such that 1. its length $n=n_An_B$,

2. dim C = (dim A)(dim B)

3. min dist C = (min dist A)(min dist B)

If, for some p_1 and p_2,

$$n_Ap_1+n_Bp_2 = 1 \bmod n_An_B,$$

then the generator polynomial of C is defined by

$$g_C(X) = GCD(g_A(X^{p_2n_B})g_B(X^{p_1n_A}), 1+X^{n_An_B})$$

EXAMPLE: Consider the (15,10) Abramson code A generated by

$$g_A(X) = (1+X)(1+X+X^4)$$

and the (7,3) Abramson code B generated by

$$g_B(X) = (1+X)(1+X+X^3).$$

Thus, $n = n_An_B = 7x15 = 105$.

Suppose we choose $p_1=1$. Then,

$$(15)(1)+(7)(p_2) = 1 \bmod 105$$

$$7p_2 = 106-15 = 91$$

$$p_2 = 13 \text{ or } -2.$$

Using $p_2=-2$, it is easily shown that

$$g_A(X^{p_2 n_B}) = X^{-70}(1+X^{14}+X^{42}+X^{70}),$$

and $$g_B(X^{p_1 n_A}) = 1+X^{30}+X^{45}+X^{60}.$$

Thus, $g_C(X) = GCD(g_A(X)g_B(X))$ may be found to be

$$g_C(X)=(1+X+X^2)(1+X^7+X^{21}+X^{35})(1+X^3+X^4)(1+X+X^2+X^3+X^4)$$
$$\times (1+X+X^2+X^4+X^6)(1+X^4+X^6+X^7+X^9+X^{10}+X^{12})$$
$$\times (1+X+X^2+X^4+X^7+X^8+X^9+X^{10}+X^{12})$$

The rate of the code is

$$R_C = 30/105 = 0.286$$

In general, the rate of this code is

$$R_C = (k_A k_B)/(n_A n_B) = R_A R_B.$$

Obviously then, the rate cannot get very close to unity; in fact, it is always lower than the code with the worse rate.

BURST ERROR CORRECTING ABILITY

Consider the cyclic product code previously described. If code A has burst error correcting ability b_A and code B can correct a burst of length b_B, then Burton and Weldon (XR4) derive the relationship for b_C, the length of burst code C

will correct, as

$$b_C \geq n_A b_B + b_A,$$

if $n_A = c_1 n_B \pm 1$, where c_1 is an integer constant.

EXAMPLE: For the (15,10) and (7,3) codes used previously,

1. $n_A = 2n_B + 1$,
2. $n_A = 15$,

and 3. $b_A = b_B = 2$ (Abramson codes).

Thus,

$$b_C \geq (15 \times 2) + 2 = 32$$

Thus, the (105,30) code C will correct bursts of length 32 or less. The authors also show that this code will correct 7 random errors.

4.9 CONCLUDING REMARKS

Several BEC codes have been studied. The Abramson codes correct only single bursts of length 2 or 3. Others, such as those in sections 3 to 8 correct long bursts with varying efficiencies. The codes of section 8 correct long bursts and random errors, but with a low rate. A decision would have to be made on which code to use depending on requirements regarding burst length and rate.

The next chapter will discuss some of the decoding procedures already known.

CHAPTER FIVE

BEC DECODING PROCEDURES

5.1 GENERAL

There are several basic methods of decoding BEC codes. One is to use shift registers only. A second is to use storage in a computer for comparison of burst patterns to received error sequences. A third is to use the high speed calculating ability of a modern computer. A fourth more common method is to combine some or all of the three previous methods.

5.2 DECODING TECHNIQUE 1

This method (XR15) involves the use of a shift register and some storage, the latter being used to store the possible burst patterns for some burst length.

Suppose V(X) is transmitted and R(X) is received. If a burst error has occurred, then

$$R(X) = V(X) + X^{i}B(X)$$

The problem is to determine both B(X) and i. The procedure follows:

1. Calculate $R_0(X) = (R(X) \bmod (1+X^n)) \bmod g(X)$
2. Calculate $R_j(X) = (XR_{j-1}(X) \bmod (1+X^n)) \bmod g(X)$ successively for $j=1,2,\ldots,n-1$ but stop when $R_j(X)$ is a recognizable pattern for the burst length in question.
3. When a recognizable pattern occurs, call it $B(X)$. Thus, $E(X)=X^{n-j}B(X)$.
4. If a recognizable pattern does not occur in n shifts, then the detected error is not correctable.

THEOREM 5.1: Let V be any (n,k) cyclic code capable of correcting every burst error $E(X)=X^iB(X)$ belonging to some set E and generated by the polynomial g(X). For some received polynomial $R(X)=V(X)+E(X)$ where V(X) is in V, let $R_0(X)=R(X) \bmod g(X)$. Then $R_0(X)$ is in E iff the error is confined to the first n-k positions.

PROOF:
1. Assume E(X) is confined to the first n-k positions. Then $R_0(X)=E(X)$ in E.
2. Assume $R_0(X)$ is in E, and further assume that E(X) is not confined to the first (n-k) positions. Then,

$$E(X) = E_1(X) + E_2(X),$$

where $E_1(X)$ is confined to the first n-k positions and $E_2(X)$ has smallest degree of at least n-k.

Let $E_2(X) \bmod g(X) = R_{02}(X)$. i.e.

$$E_2(X) = g(X)a(X) + R_{02}(X) \text{ for some } a(X)$$

$$E_1(X) + E_2(X) = E_1(X) + g(X)a(X) + R_{02}(X)$$

But $g(X)a(X)$ is a code word, and we know that a code word cannot equal the sum of correctable errors.

Thus, $E_1(X) + R_{02}(X)$ is not a correctable error.

Since $E_1(X)$ is in E, then $R_{02}(X)$ is not in E and we have a contradiction.

Therefore, $R_0(X)$ is not in E and E(X) must be confined to the first (n-k) positions.

EXAMPLE: Consider the (15,9,b=3) Abramson code with generator polynomial

$$g(X) = 1 + X^3 + X^4 + X^5 + X^6.$$

The possible burst patterns for $b \leq 3$ are

$$B(X) = \begin{bmatrix} 0 \\ 1 \\ 1+X \\ 1+X^2 \\ 1+X+X^2 \end{bmatrix}$$

Suppose $I(X)=1+X+X^3$. Then $V(X)=1+X+X^6+X^8+X^9$.

Let $E(X)=X^5(1+X+X^2)$ for example, then

$$R(X) = 1+X+X^5+X^7+X^8+X^9.$$

The steps in the procedure yield;

$$R_0(X)=R(X) \bmod g(X)=X+X^4$$

$$R_1(X)=X^2+X^5$$

$$R_2(X)=(X^3+X^6) \bmod g(X)=1+X^4+X^5$$

.

.

.

$R_{10}(X)=1+X+X^2$=a recognizable pattern.

Thus,

$$B(X) = 1+X+X^2$$

and,

$$E(X) = X^{15-10}B(X)$$

$$= X^5(1+X+X^2).$$

5.3 DECODING TECHNIQUE 2

Shiva and Zeitoun (XR18). This procedure is based partly on the principle of section 5.2 above. The main difference is that, instead of up to n decoding steps, this procedure requires approximately $\lceil k/(n-k-b+1) \rceil + 2$ steps.

Consider first the case where $n \leq 2(n-k)-b$, and suppose the error is $E(X)=X^i B(X)$, where $B(X)$ is of degree less than or equal to $b-1$. If the error is confined to the first $n-k$ digits, then $R(X) \bmod g(X)$ is the error. This accounts for errors starting at $i=0,1,2,\ldots,n-k-b$.

On the other hand, if the error is confined to the last $n-k$ digits, then

$$R^*(X) \bmod g^*(X) = E^*(X).$$

This accounts for errors starting at $i=k,k+1,\ldots n-1$. If $n \leq 2(n-k)-b$, i.e. $k \leq n-k-b$, then all positions are accounted for.

In general, for all n, the following steps are required:

0. Compute $R_0(X)=R(X) \bmod g(X)$. If $R_0(X)$ is correctable, it gives the error. Otherwise, go to step 1.
1. Let $u=(n-k-b)+1$, and let r be the smallest integer such that $ru+(n-k) \geq n$. Compute

$$R_j(X)=((X^u R_{j-1}(X)) \bmod (1+X^n)) \bmod g(X),$$

$j \geq i$. If $R_j(X)$ is correctable, then $X^{-uj}R_j(X)$ is the error. Otherwise do step j+1. If no correctable error pattern is found by step r, then the deleted burst is not correctable.

EXAMPLE: Use the same example as the previous one, i.e. with $R(X)=1+X+X^5+X^7+X^8+X^9$, where $E(X)=X^5(1+X+X^2)$. In this case $n>2(n-k)-b$ and the general procedure must be followed.

0. $R_0(X)=R(X) \bmod g(X)$
   $=X+X^4$ which is not correctable.

Now, $u=n-k-b+1=4$, and thus $r=3$. Carrying on the steps;

1. $R_1(X)=(X^4(R_0(X)) \bmod g(X)$
   $=X+X^2+X^4+X^5$ which is not correctable.
2. $R_2(X)=X^4R_1(X) \bmod g(X)$
   $=1+X+X^5$ which is not correctable.
3. $R_3(X)=X^4R_2(X) \bmod g(X)$
   $=X^2(1+X+X^2)$ which is correctable.

Thus, $E(X)=X^{-iu}R_i(X)$
$=X^{-12}R_3(X)$
$=X^{-10}(1+X+X^2)$
$=X^5(1+X+X^2)$.

5.4 DECODING OF MULTIPLE BURSTS

This procedure, developed by Tavares and Shiva (XR19), follows up section 4.6. Summarizing, binary cyclic codes were used and the errors incurred were not code words if

$$1.\quad W(E(X)) \leqslant t$$

or

$$2.\quad W(E(X)(1+X)) \leqslant t$$

where there have occurred p or less bursts, $p \leqslant t$, and $t = \lfloor (d-1)/2 \rfloor$, the number of random errors the code can correct.

Also, it was proven that the code could detect b bursts and correct p bursts each of length T_i if:

1. $b \leqslant d-1$
2. $p \leqslant b \leqslant (d-1)/2$
3. $\sum_{i=1}^{p} T_i < \max(\lfloor (3d-2b-1)/2 \rfloor, d-1)$
4. $\sum_{i=1}^{p} T_i < \max(\lfloor (3d-4p-1)/4 \rfloor, \lfloor (d-1)/2 \rfloor)$

The decoding procedure is to estimate both E(X) and E(X)(1+X) and test to decide which is the best estimate (if they differ). Suppose R(X)=V(X)+E(X). Let the decoder generate

$$R(X)(1+X) = V(X)(1+X) + E(X)(1+X).$$

Now, let § be the set of correctable burst patterns and assume that E(X) is in §. Let the estimate of E(X) from R(X) be $\hat{E}_1(X)$ and the estimate of E(X)(1+X) from R(X)(1+X) be $\hat{E}_2(X)(1+X)$. $\hat{E}_2(X)$ is obtained by dividing by (1+X). If

$$W(E(X)) \leqslant t,$$

or,

$$W(E(X)(1+X)) \leq t,$$

then at least one of $\hat{E}_1(X)$ and $\hat{E}_2(X)$ are correct if $E(X)$ is in §. Hence, if $\hat{E}_1(X)=\hat{E}_2(X)$, then

$$E(X) = \hat{E}_1(X) = \hat{E}_2(X).$$

If $\hat{E}_i(X)$ is in §, and $\hat{E}_{j,j\neq i}(X)$ is not in §, then clearly,

$$E(X) = \hat{E}_i(X).$$

If $\hat{E}_1(X)$ and $\hat{E}_2(X)$ are both in §, then the only solution is to subtract both from $R(X)$ and test to see if $R(X)-\hat{E}_i(X)$ is a code word in C. The result must be unique.

EXAMPLE: Consider the (15,7,2) BCH code generated by

$$g(X) = 1+X^4+X^6+X^7+X^8.$$

Let $I(X)=1+X^3$. Then,

$$V(X) = 1+X^3+X^4+X^6+X^8+X^9+X^{10}+X^{11}.$$

Suppose the error $E(X)=X^9(1+X)$ occurs. Then, the received polynomial will be

$$R(X) = 1+X^3+X^4+X^6+X^8+X^{11}.$$

For this code, $d=2t+1=5$. Now,

$$\sum_{i=1}^{b} T_i \leq \max(\lfloor (3d-2b-1)/2 \rfloor, d-1)$$

or $$\sum_{i=1}^{b} T_i \leq \max(7-b,4)$$

If $b=1$, $$\sum_{i=1}^{b} T_i \leq 6.$$

Thus, the code will detect one burst of length 6 or two of

length 2. Also,

$$\sum_{i=1}^{p} T_i \leq \max(\lfloor (3d-4p-1)/4 \rfloor, \lfloor (d-1)/2 \rfloor)$$

or, $$\sum_{i=1}^{p} T_i \leq \max(\lfloor (7-2p)/2 \rfloor, 2) = 2$$

Thus, the code will correct only one burst of length 2 (i.e. $p=1$, $T_1=2$).

Now, $$R(X) = 1+X^3+X^4+X^6+X^8+X^{11}$$

$$R(\alpha) = 1+\alpha^3+\alpha^4+\alpha^6+\alpha^8+\alpha^{11} = \alpha^{13}$$

The only possible errors, $R(\alpha)\neq 0$, are $\hat{E}_1(X)=X^9(1+X)$ and X^{13}.

Also, $$R(X)(1+X) = 1+X+X^3+X^5+X^6+X^7+X^8+X^9+X^{11}+X^{12}$$

$$R(\alpha)(1+\alpha) = 1+\alpha+\alpha^3+\alpha^5+\alpha^6+\alpha^7+\alpha^8+\alpha^9+\alpha^{11}+\alpha^{12} = \alpha^2$$

The only possible errors in this case are $\hat{E}_2(X)(1+X)=X^9(1+X)^2$ and $X^{13}(1+X)$. In other words, the only possibilities for $\hat{E}_2(X)$ are $X^9(1+X)$ and X^{13}.

Since $\hat{E}_1(X)=\hat{E}_2(X)$, then they are also $\hat{E}(X)$. To determine which is correct, test $\hat{V}(X)=R(X)+\hat{E}(X)$.

1. $\hat{E}(X) = X^{13}$.

   $\hat{V}(X) = 1+X^3+X^4+X^6+X^8+X^{11}+X^{13}$

   $\hat{V}(X) \bmod g(X) = 1+X^5+X^6+X^7 \neq 0$.

   Thus, X^{13} is not the error.

2. $\hat{E}(X) = X^9(1+X)$.

   $\hat{V}(X) = 1+X^3+X^4+X^6+X^8+X^9+X^{10}+X^{11}$

   $\hat{V}(X) \bmod g(X) = 0$

Therefore, the error is $X^9(1+X)$ since $V(X)$ is a code word.

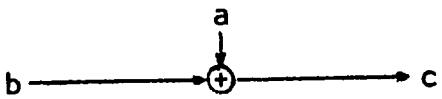
5.5 DECODING USING SEQUENTIAL CIRCUITS

Decoding procedures using shift registers have been the main method of decoding BEC codes in the past. A short discussion on how this is done follows.

5.5.1 NOTATION

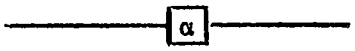
The notation of Peterson (XR10) will be used. The following three symbols are defined:

1. Adder



c=(a+b) mod 2. We will discuss only mod 2 operations.

2. Storage device



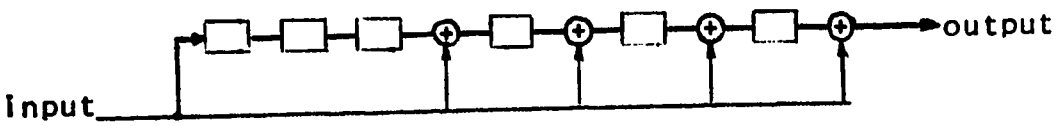
This device stores α, and thus the output is α.

3. Gate

This device is really just a relay contact. If it is closed, the circuit AB is connected, If open, AB is broken.

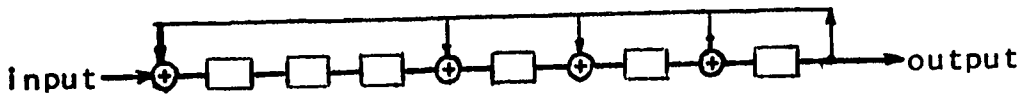
5.5.2 MULTIPLIERS AND DIVIDERS

It may easily be shown that the following circuits multiply and divide the inputs to them. The derivation may be found in Peterson (XR11).



CIRCUIT FOR MULTIPLYING BY $g(X)=1+X^3+X^4+X^5+X^6$

FIGURE 5.1



CIRCUIT FOR DIVIDING BY $g(X)=1+X^3+X^4+X^5+X^6$

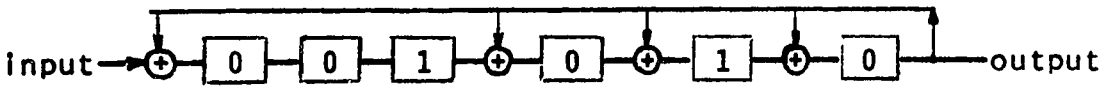
FIGURE 5.2

In both cases, g(X) is the generator polynomial for the (15,9,b=3) Abramson code ($g(X)=(1+X+X^2)(1+X+X^4)$). Also, in both cases, the input goes in highest degree first. Similarly, the first bit to reach the output is that of the highest degree.

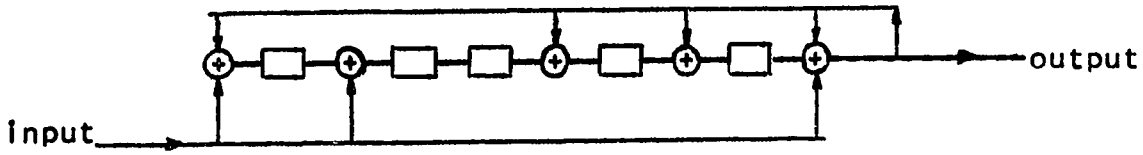
For the division case, it is useful to know that the remainder is stored in the shift register. For example, if the input is

$$1+X+X^3+X^4+X^7+X^{10}+X^{11}+X^{13},$$

then the remainder after division by g(X) is X^2+X^4. The shift register contents after division are



Simultaneous multiplication and division is also possible. Such a circuit follows:



CIRCUIT FOR MULT BY $1+X+X^5$ AND DIV BY $1+X^3+X^4+X^5+X^6$

FIGURE 5.3

Note: All registers in all three cases must have the storage devices initialized to zero.

5.5.3 CONVENTIONAL DECODER FOR A FIRE CODE

Consider the (693,676,b=d=5) Fire code generated by $g(X)=(1+X^{11})(1+X+X^6)=1+X+X^6+X^{11}+X^{12}+X^{17}$.

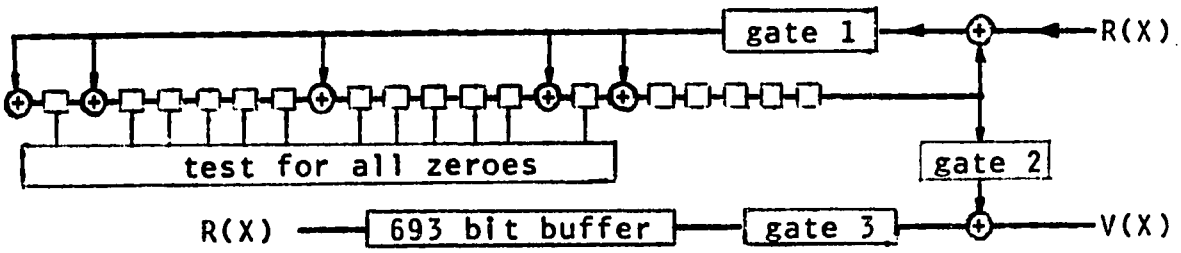


FIGURE 5.4 CIRCUIT FOR DECODING THE (693,676,b=5) FIRE CODE

Figure 5.4 shows the circuit that may be employed to decode the received polynomial R(X). It operates in the following manner.

1. Gate 1 is closed, while gates 2 and 3 are open. R(X) is fed into both the buffer and the shift register until 693 clock pulses (shifts) have occurred. R(X) is now in the buffer.
2. Gate 3 is changed to the closed position. R(X) is fed out of the buffer to the output (V(X)). At the same time, the shift register shifts with zero input until all zeroes occur in the first twelve memory units. At this time, the logic unit will detect the 'all zeroes' condition, open gate 1, and close gate 2. Also at this time, the burst of length 5 or less will be sitting in the last five memory

units.

3. The remaining bits of R(X) are added to the burst pattern, so that the output will be in the correct form (i.e. error free). If gate 2 is never closed, it means that the 'all zeroes' test failed, and that a detectable but uncorrectable error occurred.

This procedure is simple, but it requires n cycles of parity checking and n cycles of error correction. In the example above, 2x693=1386 cycles of shifting are required. This is considerably slower than it might be. If the output were fed into a modern high speed digital computer, then delays of the order of 693 bits could not be tolerated. Accordingly, the following sections are devoted to high speed decoding.

5.6 HIGH SPEED DECODING USING SEQUENTIAL CIRCUITS

5.6.1 DECODING OF FIRE CODES

An alternative decoding circuit designed by Peterson (XR16), consists of two feedback shift registers, one based on the cyclic factor $1+X^c$, and the other on the factor P(X). The two registers are run in synchronism for parity checking and error detection. Figure 5.5 shows such a circuit for the Fire

code under discussion.

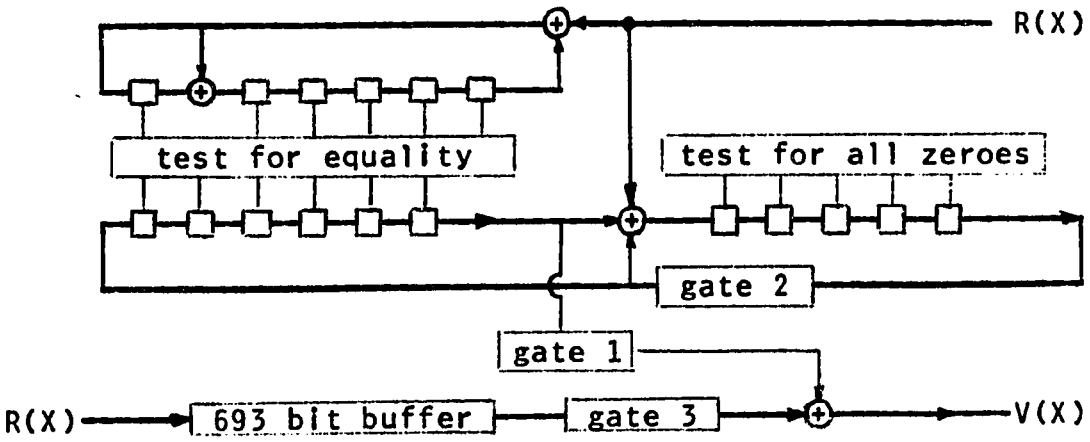


FIGURE 5.5 DECODER FOR THE (693,676,b=6) FIRE CODE

The circuit operates in the following manner.

1. Start with gates 1 and 3 open, and gate 2 closed. Feed R(X) into the shift registers and buffer as before. Close gate 3.
2. Read R(X) out of the buffer while the registers shift with zero input. Suppose a burst of length $b \leq 6$ occurs. The error may be written as $X^i B(X)$. The remainders, calculated in the registers, are;

$$s_p(X) = X^{i+6}B(X) \bmod 1+X+X^6,$$

$$\text{and, } s_c(X) = X^{i+6}B(X) \bmod 1+X^{11},$$

where p and c refer to P(X) and $(1+X^c)$ respectively. The burst will leave the buffer after n-(i+6) cycles. At that time, the contents of the registers are;

$$X^{n-i-6}s_p(X) = (X^{n-i-6})(X^{i+6}B(X) = B(X) \bmod 1+X+X^6,$$

and,

$$X^{n-i-6}s_c(X) = X^{n-i-6}X^{i+6}B(X) = B(X) \bmod 1+X^{11}.$$

Thus, if a correctable error has occurred, the two registers will each have the burst in them, and the eleven bit register will also have five zeroes. If the 'all zeroes' and equivalence tests both hold, open gate 2 and close gate 1. The sum of R(X) and the shift register contents will give the corrected code word.

3. If either of the two tests fail, then an uncorrectable error has occurred.

This circuit also takes 2x693=1386 cycles of shifting, and thus is no faster in this respect. However, the following subsection will show how to speed up the process using the circuit.

5.6.2 A FASTER METHOD

Chien (XR5) has developed a method using a combination of the previous method and some simple offline calculations.

The syndrome $s_c(X)$ is able to determine not only the burst pattern, but also the starting position up to a multiple of c. Thus, for some $0 \leq r_c < c$,

$$i = r_c \bmod c.$$

Similarly, the burst pattern is determined in the register corresponding to P(X), and for $0 \leqslant r_p < e$,

$$i = r_p \bmod e.$$

If e and c are relatively prime, then i may be determined by the Chinese Remainder Theorem. i.e.

$$i = (A_c e r_c + A_p c r_p) \bmod n$$

where A_c and A_p are integers such that

$$A_c e + A_p c = 1 \bmod n.$$

With the circuit slightly modified as shown in Figure 5.6, it should be noted that r_c is found by shifting the lower register until the 'all zeroes' test holds, and subtracting the count from c. That is,

$$r_c = c - (\text{count})_c.$$

Similarly, when the burst found in the lower register equals that in the upper register, then

$$r_p = e - (\text{count})_p.$$

One thing we know is that $c < e$. Thus,

$$(\text{count})_c \leqslant (\text{count})_p.$$

Then, simultaneous switching for decoding can occur. When the lower register stops switching (and counting), the upper register can keep switching until the burst is matched and the count is recorded.

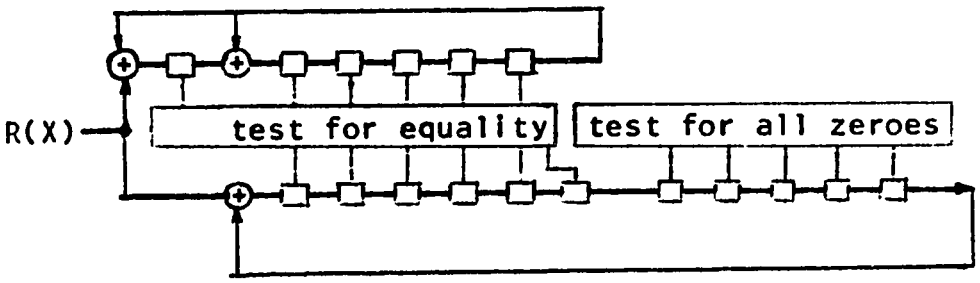


FIGURE 5.6 FAST DECODER FOR THE n=693 FIRE CODE

The counters are not shown because they work in conjunction with the clock pulse. An n bit buffer is no longer required because the burst pattern and starting position can be printed out after the received sequence.

EXAMPLE: Using the configuration of Figure 5.6, suppose the error $E(X)=X^3(1+X^4+X^5)$ occurs. The 'all zeroes' test shows equality to all zeroes at the eighth shift. Thus,

$$r_c = c-8 = 3.$$

With shifting continuing in the upper register, the match occurs at the sixtieth shift. i.e.

$$r_p = e-60 = 3.$$

In this case, then, $r_c=r_p=3$.

The computation of A_c and A_p will have been done off line and stored in the computer. In this case, $A_c=7$ and $A_p=-40$. Thus,

$$i = (A_c er_c + A_p cr_p) \bmod 693$$
$$= (7x63x3-40x11x3) \bmod 693$$
$$= 3(441-440)$$
$$= 3.$$

Since $B(X) = 1+X^4+X^5$, it follows that

$$E(X) = X^3(1+X^4+X^5).$$

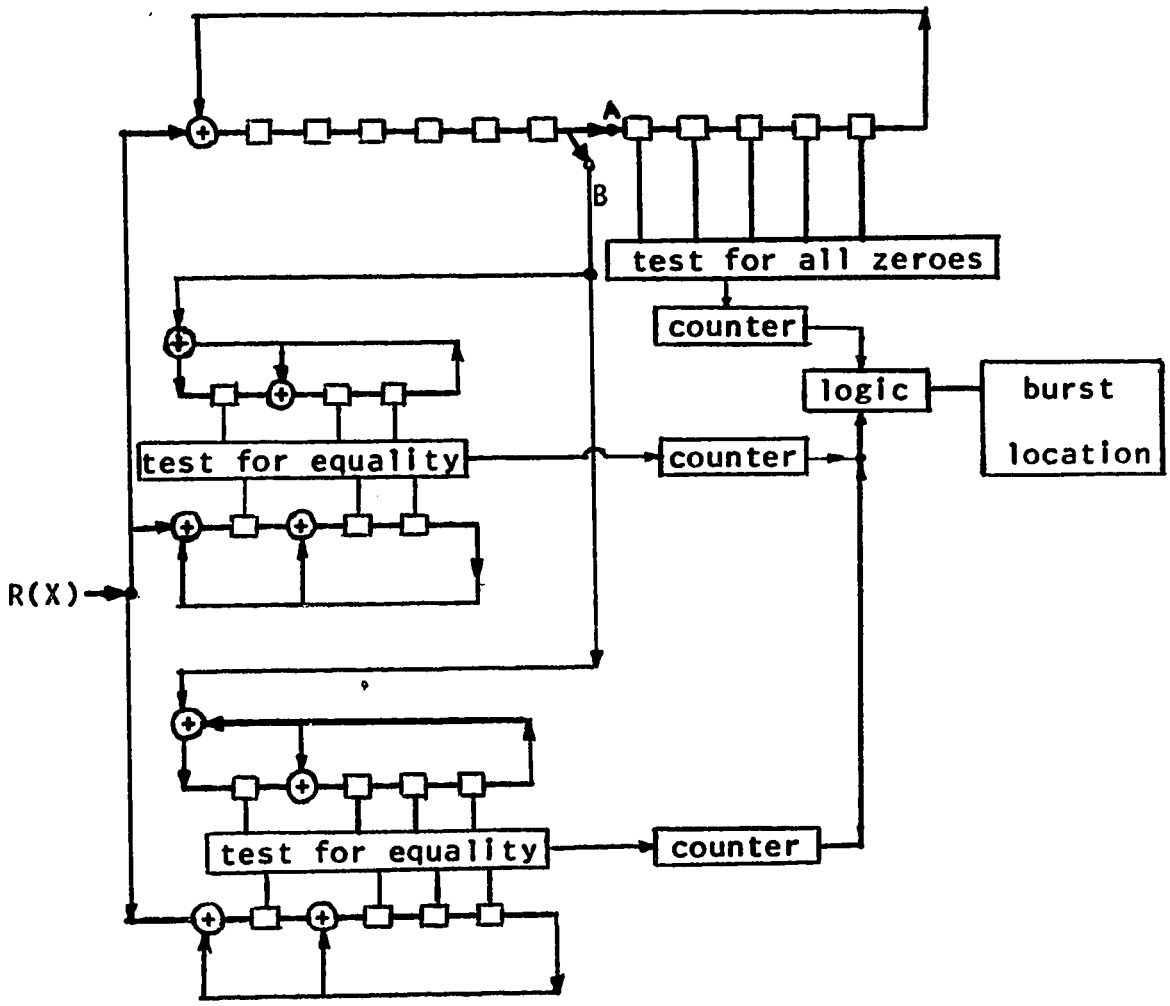
This method requires only sixty cycles of shifting while the previous methods required 693. In general, this method will improve the speed of decoding by a factor of approximately the minimum of e and c.

5.6.3 HIGH SPEED DECODING OF CHIEN CODES

Similar methods used in section 5.6.2 above may be applied to the Chien codes of section 4.4. The realizing circuit for the (1155,1137,b=6) Chien code generated by

$$g(X) = (1+X^{11})(1+X+X^4)(1+X+X^3)$$

is shown in Figure 5.7.



CIRCUIT FOR HIGH SPEED DECODING OF THE n=1155 CHIEN CODE

FIGURE 5.7

The circuit of Figure 5.7 operates in the following manner.

1. With the switch in position A, the received vector R(X) is shifted through the three registers. The burst will now be in the uppermost register. By shifting and counting, the burst may be moved such that the 'all zeroes' test applies. That is, the burst will be in the first six bits. Hence,

$$r_{11} = 11-(\text{count})_{11}.$$

Once the 'all zeroes' test has been satisfied, the two way switch is moved to B, and the three initial registers stop shifting. The burst is then simultaneously fed to the output and to the other two shift registers. Both registers then shift with zero input and counting begins. Each register stops shifting and counting when equality tests hold. In this case,

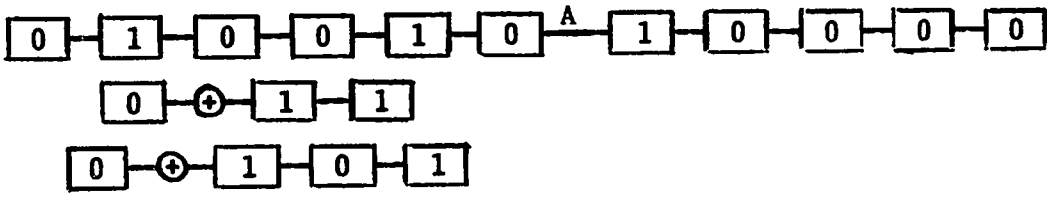
$$r_7 = (\text{count})_7,$$

and,
$$r_{15} = (\text{count})_{15}.$$

This procedure would take a maximum of (10+6+14)=30 decoding cycles versus the 1155 decoding cycles with a normal Fire code decoder of length 1155. If a burst error of length > 6 occurs, the equality test will not hold. If any register other than the uppermost one has a zero remainder, then a detected but uncorrectable error has

occurred. If all registers have zero remainders, then no error has occurred.

EXAMPLE: Suppose $E(X)=X^{12}(1+X^3+X^5)$. Then the remainders in the three shift registers will be



Thus, $(count)_{11} = 10$, and $r_{11} = 11-10 = 1$. The equality tests hold for

$$(count)_7 = 5,$$

and, $$(count)_{15} = 12.$$

Therefore, by the Chinese Remainder Theorem,

$$i=A_{11}(7x15)r_{11}+A_7(11x15)r_7+A_{15}(7x11)r_{15} \bmod 1155$$

where

$$A_{11}(7x15)+A_7(11x15)+A_{15}(7x11) = 1.$$

Substituting $r_{11}=1 \bmod 11$, $r_7=5 \bmod 7$, and $r_{15}=12 \bmod 15$ into the formula, one gets $i=12$, and thus,

$$E(X) = X^{12}(1+X^3+X^5).$$

This particular decoding operation required $10+6+12=28$ cycles of shifting, three multiplications, and a few additions and subtractions.

As a further illustration of the power of this type of decoder, consider the Fire code generated by

$$g(X) = (1+X^{37})(1+X+X^2+X^5+X^{19}).$$

This code, with e=524287 has length 19,398,619. A conventional Fire code decoder would require 19,398,619 cycles of decoding and would correct all errors of length b<19. Consider now the Chien code generated by

$$g(X) = (1+X^{37})(1+X^3+X^{10})(1+X^4+X^9).$$

This code, with the same c, e_1=1023, and e_2=511 has length n=19,341,861. This code will correct all bursts of length $\leqslant$ 9 and more than 99.6% of the bursts of lengths between 9 and 19. Thus these two codes are comparable. However, high speed decoding of the Chien code may be done in a maximum of 1077 cycles.

5.7 ONE STEP MAJORITY LOGIC DECODING

5.7.1 DEFINITION

Consider an (n,k,t) cyclic code V generated by some g(X). For an input c(X), any code word V(X) is defined by

$$V(X) = g(X)\ (c_0+c_1X+\ldots+c_{k-1}X^{k-1}).$$

Consider now a received polynomial of the form

$$R(X) = r_0+r_1X+\ldots+r_{n-1}X^{n-1}.$$

By equating powers of X, a relationship may be set up between the c_i's and the r_i's. If each c_i can be expressed in (2t+1) ways such that no r_i is repeated, then the code is said to be 1-step majority logic decodable (1SMLD).

5.7.2 THEOREM (XR17)

Let V be a binary (n,k,t) 1SMLD code generated by g(X). Then the (n'=ni,k'=ki) code V' generated by g'(X)=g(X) has the same error correcting ability as V and is 1SMLD.

The proof is given in the referenced paper.

5.7.3 THEOREM (XR17)

Any code V' above may correct by 1SMLD, the simultaneous occurrence of β bursts, each of length bi or less, and $\lfloor (d-1)/2 \rfloor - \beta b$ random errors.

PROOF: A burst of length bi can affect at most b of the parity check relations on any c_j. Thus, β such bursts will affect at most βb relations.

EXAMPLE: Consider the (15,4,3) code generated by

$$g(X) = 1+X+X^2+X^3+X^5+X^7+X^8+X^{11}.$$

If the input is $C(X)=c_0+c_1X+c_2X^2+c_3X^3$, then

$$\begin{aligned} V(X) = {} & c_0+(c_1+c_0)X+(c_2+c_1+c_0)X^2+(c_3+c_2+c_1+c_0)X^3 \\ & +(c_3+c_2+c_1)X^4+(c_3+c_2+c_0)X^5+(c_3+c_1)X^6+(c_2+c_0)X^7 \\ & +(c_3+c_1+c_0)X^8+(c_2+c_1)X^9+(c_3+c_2)X^{10}+(c_3+c_0)X^{11} \\ & +c_1X^{12}+c_2X^{13}+c_3X^{14}. \end{aligned}$$

Comparing coefficients, one gets 15 equations for the r_j's in terms of the c_i's. It is then easily shown that

$$\begin{aligned} c_0 &= r_0 \\ c_0 &= r_1+r_{12} \\ c_0 &= r_2+r_9 \\ c_0 &= r_3+r_4 \\ c_0 &= r_5+r_{10} \\ c_0 &= r_6+r_8 \\ c_0 &= r_7+r_{13} \\ c_0 &= r_{11}+r_{14} \end{aligned}$$

Here, three random errors could affect at most three parity checks on c_0 (Theory shows that the same will hold for any c_i). Alternatively, a burst of length 3, or a burst of length 2 and a random error would still affect only three equations at most. Thus, with a majority decision, c_0 may be determined.

Suppose now that i=3. That is,

$$g'(X) = 1+X^3+X^6+X^9+X^{15}+X^{21}+X^{24}+X^{33},$$

and we have a (45,12,3) code. It may easily be shown that the parity checks on c_0 now are

$$c_0 = r_0$$
$$c_0 = r_3+r_{36}$$
$$c_0 = r_6+r_{27}$$
$$c_0 = r_9+r_{12}$$
$$c_0 = r_{15}+r_{30}$$
$$c_0 = r_{18}+r_{24}$$
$$c_0 = r_{21}+r_{39}$$
$$c_0 = r_{33}+r_{42}$$

One can readily see that 3 random errors can affect at most 3 parity checks. However, a single burst of length 3 can affect at most one parity check equation. Thus the code can correct many combinations such as 1 random error and 2 bursts of length 3 or less. This illustrates the theorem of section 5.7.3.

As an extra advantage, ISMLD is extremely fast as most of the decoding is done by computer calculations. However, a high rate code with a good error correcting ability may not be found easily. For example, the (45,12,3) code used above will correct single bursts of length of up to 9 but has a rate of only R=0.267.

CHAPTER SIX

A NEW DECODING PROCEDURE

6.1 INTRODUCTION

Consider an (n,k) cyclic code V' generated by g'(X) and capable of correcting all bursts of length b or less. For a code word V'(X), and burst error pattern B(X) starting in position i+1, we have the received polynomial R(X) defined by

$$R'(X) = V'(X) + X^{i}B(X).$$

In section 5.2 it was shown that the decoding of R(X) could be accomplished by examination of

$$R_j(X) = ((X^j R_{j-1}(X)) \bmod (1+X^n)) \bmod g(X)$$

for $1 \leqslant j \leqslant k+b$ and given that $R_0(X)=R(X) \bmod g(X)$. This method could require up to k+b steps of decoding.

In section 5.3, it was shown that R(X) need only be shifted u=n-k-b+1 bits at a time, and thus approximately $\lfloor k/u \rfloor$ +2 steps are required. However, for a high rate code, this does not represent a significant reduction. For example, the (1155,1137,b$\leqslant$6) Chien code discussed earlier could be decoded in approximately 89 steps compared to 1143 steps in section 5.2. Thus the reduction is only a factor of approximately u.

For both techniques, the number of shifts required is large, but neither has the disadvantageous requirement of storing syndromes or doing off line calculations. In such long code situations, Chien's procedure is very useful. While it requires only a small amount of computation as well as shifting, it requires the generator polynomial to have $(1+X^c)$ as a factor. Fire codes have the same restriction.

The decoding technique presented in this chapter does not require a code which contains $(1+X^c)$ as a factor of $g(X)$, but it does require that $g(X)$ be self-reciprocal. The decoding procedure involves very little shifting, some calculations, and some storage of syndromes. However, the storage required is small if b is appropriately small in the context of storage space available. That is, only 2^{b-1} syndromes need be stored.

6.2 A SUBCODE V OF V'

Consider a set V which is a subset of V', such that V is the code generated by

$$g(X) = LCM(g'(X), g'^{*}(X))$$

where * is the reciprocal as previously defined. Suppose

$$g'(X) = g_1(X)g_2(X)\ldots g_r(X)$$

where each $g_j(X)$ is irreducible. Further, let α^{a_j} be a root of $g_j(X)$, where α is a primitive element of $GF(2^m)$, m being the

smallest integer such that n divides 2^m-1. Since g(X) = LCM(g'(X),g'*(X)), α^{-a_j} is also a root of g(X).

6.3 CALCULATING THE LENGTH OF THE BURST

Consider the received polynomial

$$R(X) = V(X) + X^i B(X)$$

where V(X) is in V and B(X) is a burst of length $\ell \leq b$. Consider now, the product $R(X)R^*(X)$.

6.3.1 RESULT

There exists a $\theta \leq b$ such that

$$(X^\theta R(X)R^*(X) \bmod (1+X^n)) \bmod g(X) = B(X)B^*(X).$$

This may be shown in the following manner. We know that

$$\text{LHS} = X^\theta R(X)R^*(X) \bmod (1+X^n) \bmod g(X)$$
$$= X^\theta E(X)E^*(X) \bmod (1+X^n) \bmod g(X).$$

But $E(X)=X^i B(X)$ and thus $E^*(X)=X^{n-i-\ell}B^*(X)$. Therefore,

$$\text{LHS} = X^\theta X^{n-\ell} B(X)B^*(X) \bmod (1+X^n) \bmod g(X)$$
$$= X^{n-\ell+\theta} B(X)B^*(X) \bmod (1+X^n) \bmod g(X)$$
$$= X^{\theta-\ell} B(X)B^*(X) \bmod (1+X^n) \bmod G(X).$$

Suppose $\theta=0$. Then $\text{LHS}=X^{-\ell}B(X)B^*(X)$ which is not the RHS. As θ is increased, the first chance for equality is when $\theta=\ell$. If

equality does not occur at this point, then it is increased until the boundary defined by g(X) is reached. The cyclic nature of the parent code then dictates that equality must be reached if the error is correctable.

6.3.2 CALCULATIONS

B(X) has degree $\ell-1 \leq b-1$, and $B^*(X)$ has degree $\ell-1 \leq b-1$ also. Thus $B(X)B^*(X)$ has degree $2\ell-2 \leq 2b-2$. Thus, for some ψ, when $X^{\theta}R(X)R^*(X) \bmod (1+X^n) \bmod g(X)$ has degree $\psi \leq 2b-1$, then, $\psi=2\ell-2$, or

$$\ell = (\psi+2)/2.$$

EXAMPLE: If $(X^{\theta}R(X)R^*(X) \bmod (1+X^n)) \bmod g(X) = 1+X^2+X^4+X^6$, and $n-k>6$, then $\psi=6$, and $l=(6+2)/2=4$.

Knowing the length of the burst will not be absolutely necessary, but useful in the following sections.

6.4 DECODING THEORY

We know from section 6.2 that α^{a_j} and α^{-a_j} are roots of g(X). Thus,

$$R(\alpha^{a_j}) = V(\alpha^{a_j}) + \alpha^{ia_j}B(\alpha^{a_j})$$

$$= \alpha^{ia_j} B(\alpha^{a_j}) \qquad j=1,\ldots,r$$

Similarly,

$$R(\alpha^{-a_j}) = \alpha^{-ia_j} B(\alpha^{-a_j}) \qquad j=1,\ldots,r$$

Therefore,

$$R(\alpha^{a_j})R(\alpha^{-a_j}) = \alpha^{ia_j} B(\alpha^{a_j})\alpha^{-ia_j} B(\alpha^{-a_j})$$

$$= B(\alpha^{a_j})B(\alpha^{-a_j}) \qquad j=1,\ldots,r$$

However,

$$B(\alpha^{-a_j}) = \alpha^{-(\ell-1)a_j} B^*(\alpha^{a_j})$$

where ℓ is the length of $B(X)$. It follows then, that

$$R(\alpha^{a_j})R(\alpha^{-a_j}) = B(\alpha^{a_j})\alpha^{-(\ell-1)a_j} B^*(\alpha^{a_j})$$

$$= \alpha^{-(\ell-1)a_j} B(\alpha^{a_j}) B^*(\alpha^{a_j}) \qquad j=1,\ldots,r$$

Let us now use the notation R_j where

$$R_j = R(\alpha^{a_j})R(\alpha^{-a_j}).$$

That is,

$$R_j = \alpha^{-(\ell-1)a_j} B(\alpha^{a_j}) B^*(\alpha^{a_j}), \qquad j=1,\ldots,r.$$

Suppose R_j is computed for all distinct bursts $B_k(X)$. Now, any given R_j may not have distinct values for all values of k. For instance, bursts of equal length that are reciprocals will give the same R_j. Thus, we store only distinct syndromes, which we shall call s_k defined by

$$s_k = (D_{k1}(\alpha), D_{k2}(\alpha), \ldots, D_{kr}(\alpha)),$$

where

$$D_{kj}(\alpha) = \alpha^{-(\ell_k-1)a_j} B_k(\alpha^{a_j}) B_k^*(\alpha^{a_j}),$$

and where the s_k are all distinct.

If $R(\alpha^{a_j})R(\alpha^{-a_j})$ is calculated for all $j=1,\ldots,r$ and

compared to the s_k syndromes, then the choice of possible burst patterns is narrowed down. Having calculated ℓ previously also narrows the choice. For example, if ℓ is not more than 6, we know that not more than two choices exist.

6.4.1 RESULT

There is only one B(X) such that

$$(R(\alpha^{a_i})B^{-1}(\alpha^{a_i}))^{1/a_i}$$

is equal to the same quantity, say α^s for i=1,...,r.

PROOF: Suppose there are two patterns $B_1(X)$ and $B_2(X)$ such that $(R(\alpha^{a_i})B_1^{-1}(\alpha^{a_i}))^{1/a_i} = \alpha^{s_1}$ i=1,...,r ...A

and, $(R(\alpha^{a_i})B_2^{-1}(\alpha^{a_i}))^{1/a_i} = \alpha^{s_2}$ i=1,...,r ...B

Then, from A, we have $R(\alpha^{a_i}) = \alpha^{a_i s_1} B_1(\alpha^{a_i})$,

and from B, we have $R(\alpha^{a_i}) = \alpha^{a_i s_2} B_2(\alpha^{a_i})$.

This means that

$$\alpha^{a_i s_1} B_1(\alpha^{a_i}) = \alpha^{a_i s_2} B_2(\alpha^{a_i}) \text{ for } i=1,...,r.$$

In other words, the syndrome

$$(R(\alpha^{a_1}), R(\alpha^{a_2}), ..., R(\alpha^{a_r}))$$

is the same for two errors.

But this is impossible in view of the error correcting capability of the parent code. This means that there cannot be two patterns $B_1(X)$ and $B_2(X)$ satisfying both A and B above.

6.5 DECODING PROCEDURE

The decoding may be done in three or four steps.

1. Calculate $(R(X)R^*(X) \bmod (1+X^n)) \bmod g(X)$ and shift mod g(X) until the degree ψ is $\leqslant 2b-2$. Calculate ℓ from the formula $\ell=(\psi+2)/2$.
2. Compute $R(\alpha^{a_j})R(\alpha^{-a_j})$, $j=1,\ldots,r$ from the received polynomial. This will give the required syndrome.
3. Compare the computed syndromes with the stored syndromes and obtain the possible burst patterns. Narrow the number of patterns down by using the calculated value ℓ. If only one burst pattern applies, then
$$\alpha^i = (R(\alpha^{a_1})B^{-1}(\alpha^{a_1}))^{1/a_1}.$$
4. If two or more bursts are possible, test one of them with
$$\alpha^i = (R(\alpha^{a_j})B^{-1}(\alpha^{a_j}))^{1/a_j} \text{ for all } j.$$
If the same value of i is obtained for all j, then that burst pattern is the error pattern and i is the starting position. Otherwise, if failure occurs for even one j, then the second pattern must be tested in a similar manner and so on. When the correct pattern is determined, then i may be found by
$$\alpha^i = (R(\alpha^{a_1})B^{-1}(\alpha^{a_1}))^{1/a_1}.$$

NOTES: 1. It is possible for $B(\alpha^{a_j})=0$ for some j. Thus computation of $B^{-1}(\alpha^{a_j})$ is not possible and the computer program must be set up to test for this condition. When such a case occurs, simply test other bursts first. It might be the correct burst by default!

2. The most number of tests required in step 4 is r. Normally, not more than two tests and one calculation are required.

6.6 RATE

Fire codes have some of the highest rates of codes discussed in Chapter 4. The codes discussed in this thesis may be a subset of Fire codes and, for long lengths, will have approximately the same rates.

For example, consider the $(693,676,b\leq 6)$ Fire code generated by

$$g'(X) = (1+X^{11})(1+X+X^6).$$

This code has the rate R=0.975. The sub-code of this would be a $(693,670,b\leq 6)$ code generated by

$$g(X) = \text{LCM}\ (\ (1+X^{11})(1+X+X^6), (1+X^{11})(1+X^5+X^6)\ )$$
$$= (1+X^{11})(1+X+X^6)(1+X^5+X^6).$$

This code has rate R=0.967 which is not much different.

Since an (n,k,b) Fire code becomes an (n,k-b,b) code, it is obvious that for high k, the rate of a Fire code will not decrease appreciably. That is, the difference between the rates of the two codes is

$$R_F - R = k/n - (k-b)/n$$
$$= (k-k+b)/n$$
$$= b/n.$$

If $b \ll n$, then R_F is approximately R.

A comparison of rates of different codes with the sub-code presented in this thesis in shown in Table 6.1.

	EXISTING CODE				THESIS SUB-CODE			
Type	n	k	b	Rate	n	k	b	Rate
Fire	105	94	4	0.895	105	90	4	0.857
	315	304	3	0.965	315	301	3	0.956
	693	676	6	0.975	693	670	6	0.967
	1785	1770	4	0.992	1785	1766	4	0.989
Chien	1155	1137	6	0.984	1155	1130	6	0.978
	2821	2800	7	0.993	2821	2792	7	0.990
	7905	7879	9	0.997	7901	7870	9	0.996
	11067	11040	9	0.998	11067	11030	9	0.997
	61845	61814	10	0.999	61845	61802	10	0.999
Abramson	63	56	2	0.889	63	50	2	0.794
	63	55	3	0.873	63	49	3	0.778
Gorog S_G	189	177	4	0.937	189	171	4	0.904
	315	299	6	0.949	315	293	6	0.930

COMPARISON OF THE RATES OF EXISTING CODES

WITH THE THESIS SUB-CODES

TABLE 6.1

Table 6.1 demonstrates that the sub-code presented here is only slightly less efficient than existing burst error correction codes as discussed in Chapter 4.

6.7 DECODING SPEED

Rate is not the only consideration. Ease of decoding is important, especially when discussing codes relative to Chien codes.

Table 4.2 showed some Chien codes that required from 30 to 160 cycles of decoding and 5 to 7 multiplications to correct burst lengths of from 6 to 12. This method is by far the fastest decoding method previously described. Suppose we consider the (1155,1137,b<6) Chien code generated by

$$g(X) = (1+X^{11})(1+X+X^4)(1+X+X^3).$$

This code requires 30 cycles of decoding and 5 multiplications.

Consider now the (1155,1130,b≤6) sub-code generated by

$$g(X) = (1+X^{11})(1+X+X^4)(1+X^3+X^4)(1+X+X^3)(1+X^2+X^3).$$

This requires a computer multiplication, six register shifts, one comparison, and up to four calculations.

It is difficult to say which type of decoding procedure is more efficient. However, one would assume that the Chien procedure has a slight edge. It must be considered, however,

that for a code not of the Chien type and at the same time reciprocal, then the decoding procedure presented in this thesis is most likely the easiest to use provided that storage space is readily available.

6.8 WORKED EXAMPLE

Consider the (63,44,b$\leq$4) Fire code generated by

$$g'(X) = (1+X^7)(1+X+X^6).$$

The sub-code then would have generator polynomial

$$g(X) = \text{LCM}\ (\ g'(X), g'^*(X)\ )$$
$$= 1+X+X^5+X^6+X^8+X^{11}+X^{13}+X^{14}+X^{18}+X^{19}.$$ This code will be calculated on GF(2^6). Suppose I(X)=(1+X). Then,

$$V(X) = 1+X^2+X^5+X^7+X^8+X^9+X^{11}+X^{12}+X^{13}+X^{15}+X^{18}+X^{20}.$$

and suppose that $E(X) = X^{17}(1+X+X^3)$. Therefore,

$$R(X) = 1+X^2+X^5+X^7+X^8+X^9+X^{11}+X^{12}+X^{13}+X^{15}+X^{17},$$

and

$$R^*(X) = X^{46}(1+X^2+X^4+X^5+X^6+X^8+X^9+X^{10}+X^{12}+X^{15}+X^{17}).$$

Step 1. $R(X)R^*(X) \bmod 1+X^{63} = 1+X^2+X^3+X^5+X^6+X^9+X^{10}+X^{17}+X^{46}$
$+X^{53}+X^{54}+X^{57}+X^{58}+X^{60}+X^{61}.$

$R_0 = RR^* \bmod (1+X^{63}) \bmod g(X) = X+X^4+X^7+X^8+X^{11}+X^{12}$
$+X^{13}+X^{15}+X^{16}+X^{17}+X^{18}.$

$R_1 = XR_0 \bmod g(X) = 1+X+X^2+X^6+X^9+X^{11}+X^{12}+X^{16}+X^{17}.$

$R_3 = X^2R_1 \bmod g(X) = 1+X+X^2+X^3+X^4+X^5+X^6$, which has

$6 < n-k$. Thus $\psi=6$ and $\ell=(6+2)/2=4=b$.

i.e. $\ell=4$.

Step 2. The roots of $g'(X)$ are $a_1=1$ and $a_2=9$, where α is the root of $1+X+X^6$ and α^9 is the root of $1+X^7$.

Note: At this point it is useful to note that for

$$g(X) = g_1(X)g_2(X)\ldots g_r(X),$$

$R(\alpha^{a_j})$ may be found by calculating $R(X)$ mod $g_j(X)$ at the point $X=\alpha^{a_j}$.

Case 1: Root α^{a_1}. $R(\alpha^{a_1}) = R(\alpha) = \alpha^{52}$.

$R(\alpha^{-a_1}) = R(\alpha^{-1}) = \alpha^{28}$.

Thus $R(\alpha^{a_1})R(\alpha^{-a_1}) = \alpha^{52}\alpha^{28} = \alpha^{17}$.

Case 2: Root α^{a_2}. $R(\alpha^{a_2}) = R(\alpha^9) = \alpha\alpha^{18}$.

$R(\alpha^{-a_2}) = R(\alpha^{-9}) = 0$.

Thus $R(\alpha^{a_2})R(\alpha^{-a_2}) = 0$.

Therefore, the calculated syndrome is $(\alpha^{17},0)$.

Step 3. The stored syndromes for the possible bursts follow in Table 6.2.

	BURST PATTERN	SYNDROME	
	1	$(1,1)$	
	$1+X$	$(\alpha^{11},\alpha^{14})$	
	$1+X^2$	$(\alpha^{22},\alpha^{36})$	
	$1+X+X^2$	$(\alpha^{50},\alpha^{54})$	
	$1+X^3$	(α^{61},α^{9})	
*****	$1+X+X^3, 1+X^2+X^3$	$(\alpha^{17},0)$	*****
	$1+X+X^2+X^3$	$(\alpha^{33},\alpha^{54})$	

TABLE 6.2 STORED SYNDROMES FOR GIVEN BURSTS

The computed syndrome matches the starred syndrome above, indicating that $B_1(X)=1+X+X^3$ or $B_2(X)=1+X^2+X^3$.

Step 4. Test $B_1(X)=1+X+X^3$. With a_1, $\alpha^i=\alpha^{17}$. With a_2, $\alpha^i=\alpha^{3,10,17,24,31,38,45,52,59}$. Intersection occurs for α^{17}. Thus it is not necessary to test the other pattern.

Therefore, $E(X) = X^{17}(1+X+X^3)$.

CHAPTER SEVEN

CONCLUDING REMARKS

This thesis has, in Chapter Two, attempted to lay an algebraic base for the coding theory that followed in Chapter Three initially and the other chapters as well. Not all algebraic concepts were covered. An attempt was made to introduce only those which were necessary to develop the coding and decoding methods for burst error correction.

Chapter Three was a very brief discussion of binary cyclic codes. Again, the concepts used were only the ones necessary to the rest of the thesis.

Chapter Four was a thorough discussion of the theory of burst error correcting codes and the major existing BEC codes. Wherever possible, the codes' burst error correcting abilities and rates were compared in an attempt to determine which codes were superior and under which circumstances.

Existing burst error correction decoding procedures were discussed in Chapter Five. Again, the decoding procedures were compared for efficiency. The main difference in decoding comparison was that speed and ease of decoding were the

guidelines. One would have to conclude that the Chien codes discussed in Section 5.6.3 were the easiest to decode when using Chien's procedure.

In Chapter Six, a new decoding procedure was introduced using, in most cases, a sub-code of any existing code. It was shown in detail in Table 6.1 that the rate of the thesis sub-code was generally nearly as good as that of the parent code. Further, discussion showed that the speed of decoding was, with the possible exception of the Chien procedure, much easier and faster. The advantage of this method over that of Chien was that the latter code required a generator polynomial that had in it a factor $(1+X^c)$. Since this restriction is not placed on the sub-code discussed, it may be concluded that under certain conditions it is more advantageous to use the sub-code presented and its associated decoding procedure.

BIBLIOGRAPHY

1. Abramson, N.M., 'A Class of Systematic Codes for Non-Independent Errors', IRE Trans., IT-5, 150-157 (1959).
2. Bahl, L.R. and R.T. Chien, 'Multiple-Burst-Error Correction by Threshold Decoding', Inf and Control, 15, 397-406, (1969).
3. Bose, R.C., and D.K. Ray-Chaudhuri, 'On a Class of Error Correcting Binary Group Codes', Inf. and Control, 3, 68-79 (1960).
4. Burton, H.O. and E.J. Weldon, Jr., 'Cyclic Product Codes', IEEE Trans on IT, Vol.IT-11, 3, 433-439 (1965).
5. Chien, R.T., 'Burst-Correcting Codes with High-Speed Decoding', IEEE Trans on IT, Vol. IT-15, 1, 109-113(1969).
6. Fire, P., 'A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors', Sylvania Report RSL-E-2, Sylvania Reconnaissance Systems Laboratory, Mountain View, California (1959).
7. Gorog, E., 'Some New Classes of Cyclic Codes Used for Burst-Error Correction', IBM Journal, April 1963, 102-111.
8. Hocquenghem, A., 'Codes Correcteurs d'Erreurs', Chiffres, 2, 147-156, (1959).
9. Melas, M., 'A New Group of Codes for Correction of

Dependent Errors in Data Transmission', IBM Journal, 4, 58 (1960).

10. Peterson, W.W., 'Error-Correcting Codes', M.I.T Press, (1961), 107-136.
11. op. cit. 108-114.
12. op. cit. 162-182.
13. op. cit. 169-180.
14. op. cit. 183.
15. op. cit. 189-191.
16. op. cit. 196-198.
17. Shiva, S.G.S. and R. Provost, 'On 1-Step Majority Logic Decodable Binary Block Codes', EE5631 lecture notes by S.G.S. Shiva (1970), University of Ottawa, Ottawa, Canada.
18. Shiva, S.G.S. and T. Zeitoun, 'Decoding of Binary Cyclic Burst-Error-Correcting Codes', IEEE Trans on IT, Nov 1969, Vol 15, 6, 737.
19. Tavares, S.E. and S.G.S. Shiva, 'Detecting and Correcting Multiple Bursts for Binary Cyclic Codes', IEEE Trans on IT, Sep 1970, Vol 16, 5, 643.
20. Tavares, S.E., S.G.S. Shiva, and P.R. McIntyre, 'A Class of Cyclic Codes for Correcting Single Bursts', Digest of the 1970 Canadian Symposium on Communications, Montreal, Canada, Nov 12-13, (1970).

VITAE

Full Name:	Peter Ramsay McINTYRE
Birth Place:	Kirkland Lake, Ontario
Birth Date:	January 12, 1943
High Schools:	University of Toronto Schools
	Leaside High School
Baccalaureate	Queen's University at Kingston (BSc(Eng))